

SISTEMA PROGRAMABLE MICRO CONTROLADO PARA EL CONTROL DE UNA AERONAVE DE DESPEGUE VERTICAL Y VUELO HORIZONTAL

Santiago Miranda Franco
Luis Gabriel Martínez Ordóñez

Universidad tecnológica de Pereira
Programa de Ingeniería Mecatrónica
Facultad de Tecnologías
Noviembre de 2020

SISTEMA PROGRAMABLE MICRO CONTROLADO PARA EL CONTROL DE
UNA AERONAVE DE DESPEGUE VERTICAL Y VUELO HORIZONTAL

Santiago Miranda Franco.

C.C 1088351870

Luis Gabriel Martínez Ordóñez.

C.C 87062834

Director.

Adonái Zapata Gordon

Proyecto de Grado.

Tecnólogo Mecatrónico

Universidad Tecnológica de Pereira

Programa de Ingeniería Mecatrónica

Facultad de Tecnologías

Noviembre de 2020

ÍNDICE GENERAL

DEFINICIÓN DE LA PROPUESTA	7
1.1 INTRODUCCIÓN	7
1.2 PROBLEMA DE INVESTIGACIÓN	8
1.3 DELIMITACIÓN	8
1.4 OBJETIVOS.....	9
1.4.1 <i>Objetivo General</i>	9
1.4.2 <i>Objetivos específicos</i>	9
1.5 JUSTIFICACIÓN	9
VEHÍCULOS AÉREOS NO TRIPULADOS	11
2.1. CLASIFICACIÓN	11
2.2. AERONAVES DE DESPEGUE VERTICAL Y VUELO ORIZONTAL (VTOL)	12
2.3. SISTEMA AUTOPILOTO	13
2.3.1. <i>Clasificación</i>	13
2.4. PIXHAWK 2.4.8	20
2.4.1. <i>Características principales</i>	20
2.5. SOFTWARE OPEN SOURCE	22
2.5.1. <i>Software de programación y de control</i>	23
2.5.2. <i>Software de simulación</i>	25
DISEÑO Y CONSTRUCCIÓN DEL SISTEMA AUTOPILOTO	30
3.1. MECÁNICA Y ELECTROMECAÁNICA DE LA AERONAVE	30
3.2. ELECTRÓNICA DE LA AERONAVE	31
3.3. COMUNICACIONES	34
3.4. ENSAMBLE.....	35
3.5. AUTOPILOTO DE LA AERONAVE	39
3.6. PROGRAMACIÓN, SIMULACIÓN Y CALIBRACIÓN DEL AUTOPILOTO	42
ENTRENAMIENTO MEDIANTE SOFTWARE DE AEROMODELISMO	53
4.1. CONFIGURACIÓN DEL JOYSTICK	53
4.1.1. <i>Instalación del Launcher para la detección del mando</i>	53
4.1.2. <i>Configuración de la aplicación y conexión al servidor</i>	54
4.1.3. <i>Configuración del mando en el software de simulación</i>	55
4.2. ELECCIÓN DE LA AERONAVE	58
4.3. ACTIVACIÓN DE LOS BINOCULARES	60
4.4. DESARROLLO DE SIMULACIONES.....	60
RESULTADOS OBTENIDOS (VALIDACIÓN DEL FUNCIONAMIENTO)	62
5.1. PRUEBAS PRELIMINARES	62
5.1.1. <i>Transmisión de datos</i>	62
5.1.2. <i>Vista de vuelo</i>	63
5.1.3. <i>Testeo de los motores</i>	64
5.2. PRUEBA DE LA AERONAVE	65
5.2.1. <i>Posicionamiento y armado de la aeronave</i>	65
6. PRESUPUESTO	67
7. CONCLUSIONES	70
8. RECOMENDACIONES	71
9. BIBLIOGRAFÍA.....	72
10. ANEXOS	75

LISTA DE FIGURAS

FIGURA 1.1 AERONAVE TIPO VTOL FUSIÓN.....	8
FIGURA 2.1 UAV DE RECONOCIMIENTO KORSAR DE ORIGEN RUSSO	12
FIGURA 2.2 UAV DE COMBATE GENERAL ATOMICS MQ-9 REAPER.....	12
FIGURA 2.3 DJI PHANTOM 4 Pro V2.0.....	12
FIGURA 2.4 SISTEMA AUTOPILOTO iNAVY Y SOFTWARE DE CONTROL.....	14
FIGURA 2.5 PLACA DE CONTROL NAZE32	14
FIGURA 2.6 PLACA DE CONTROL OPENPILOT CC3D EVO	15
FIGURA 2.7 TARJETA DE CONTROL LIBREPILOT Y SOFTWARE DE MANEJO	15
FIGURA 2.8 PLACA DE CONTROL NAZA DJI	16
FIGURA 2.9 PLACA DE CONTROL ARDUPILOT	16
FIGURA 2.10 PLACA DE CONTROL Y SOFTWARE DE BETAFLIGHT	17
FIGURA 2.11 PLACA DE CONTROL CLEANFLIGHTV	18
FIGURA 2.12 PLACA DE CONTROL KISS.....	19
FIGURA 2.13 PLACA DE CONTROL MULTIWII.....	19
FIGURA 2.14 PLACA DE CONTROL TAU LABS	20
FIGURA 2.15 PLACA DE CONTROL PIXHAWK 2.4.8 E INDICADORES DE PUERTOS DE CONEXIÓN.....	20
FIGURA 2.16 INTERFAZ DEL PROGRAMA APM PLANNER 2.0.....	23
FIGURA 2.17 INTERFAZ DEL PROGRAMA QGROUND CONTROL	24
FIGURA 2.18 INTERFAZ DEL PROGRAMA MISSION PLANNER	25
FIGURA 2.19 VISUALIZACIÓN DEL ENTORNO DE SIMULACIÓN DE JMAVSIM	26
FIGURA 2.20 VISUALIZACIÓN DEL ENTORNO DE SIMULACIÓN DE FLIGHTGEAR	26
FIGURA 2.21 EJEMPLO DE VEHÍCULO TERRESTRE SIMULADO EN GAZEBO	27
FIGURA 2.22 EJEMPLO DE VTOL SIMULADO EN GAZEBO	27
FIGURA 2.23 EJEMPLO DE ENTORNO DE SIMULACIÓN EN AIRSIM	28
FIGURA 2.24 PANTALLA DE CARGA Y MENÚ DE INICIO DEL PROGRAMA [LOS AUTORES].....	28
FIGURA 2.25 PANTALLA DE CARGA DE DROIDJOY APP Y MENÚ PRINCIPAL DEL SERVIDOR [LOS AUTORES].....	29
FIGURA 3.1 PIEZAS DEL CHASIS DJI F450 [LOS AUTORES]	30
FIGURA 3.2 ENSAMBLE DE LA ESTRUCTURA DJI F450 [LOS AUTORES]	30
FIGURA 3.3 MOTORES BRUSHLESS OUTRUNNER 2212 920KV [LOS AUTORES].....	31
FIGURA 3.4 HÉLICES DJI GENÉRICAS F450 [LOS AUTORES].....	31
FIGURA 3.5 ESC'S IMPLEMENTADOS EN LA AERONAVE [LOS AUTORES].....	32
FIGURA 3.6 BATERÍA ZIPPY 2200 MAH [LOS AUTORES].....	33
FIGURA 3.7 MÓDULO DE ALIMENTACIÓN 3DR [LOS AUTORES].....	33
FIGURA 3.8 MÓDULO GPS Y ESTRUCTURA DE SOPORTE [LOS AUTORES]	34
FIGURA 3.9 MÓDULO DE TELEMETRÍA [LOS AUTORES].....	34
FIGURA 3.10 PPM ENCODER [LOS AUTORES]	35
FIGURA 3.11 CONTROL REMOTO FLYSKY FS-T6 Y RECEPTOR RB-6 [LOS AUTORES]	35
FIGURA 3.12 FIJACIÓN DEL MÓDULO GPS [LOS AUTORES]	36
FIGURA 3.13 ACOPLE DEL SISTEMA AUTOPILOTO [LOS AUTORES]	36
FIGURA 3.14 SENTIDO DE GIRO DE LOS MOTORES [LOS AUTORES]	37
FIGURA 3.15 ACOPLE DE LOS MOTORES BRUSHLESS [LOS AUTORES].....	37
FIGURA 3.16 ACOPLE DE LOS ESC'S [LOS AUTORES]	37
FIGURA 3.17 ACOPLE DEL BUZZER Y DEL INTERRUPTOR DE SEGURIDAD [LOS AUTORES]	38
FIGURA 3.18 ACOPLE DEL MÓDULO DE TELEMETRÍA [LOS AUTORES]	38
FIGURA 3.19 FIJACIÓN DEL PPM ENCODER Y EL RECEPTOR RC [LOS AUTORES]	38
FIGURA 3.20 UBICACIÓN DE LA BATERÍA [LOS AUTORES].....	39
FIGURA 3.21 PUERTOS DE CONEXIÓN DEL INTERRUPTOR DE SEGURIDAD Y EL BUZZER [LOS AUTORES].....	39
FIGURA 3.22 CONEXIÓN DEL MÓDULO GPS [LOS AUTORES]	40
FIGURA 3.23 CONEXIÓN DEL MÓDULO DE TELEMETRÍA [LOS AUTORES]	40
FIGURA 3.24 CONEXIÓN DE LOS ESC [LOS AUTORES]	40
FIGURA 3.25 CONEXIÓN Y PRUEBA DE LOS MOTORES [LOS AUTORES].....	41
FIGURA 3.26 CONEXIÓN DE PINES DE CONTROL DE LOS ESC'S [LOS AUTORES]	41
FIGURA 3.27 CONEXIÓN DEL RECEPTOR R/C AL PPM ENCODER [LOS AUTORES]	42
FIGURA 3.28 CONEXIÓN DEL PPM ENCODER AL SISTEMA AUTOPILOTO [LOS AUTORES].....	42
FIGURA 3.29 SELECCIÓN Y CARGA DEL FIRMWARE [LOS AUTORES].....	43
FIGURA 3.30 PROCESO DE CARGA DEL FIRMWARE Y VERIFICACIÓN [LOS AUTORES]	43
FIGURA 3.31 ELECCIÓN DE LA ESTRUCTURA DE LA AERONAVE [LOS AUTORES]	44
FIGURA 3.32 SUMARIO DE PARÁMETROS CONFIGURADOS Y RESTANTES [LOS AUTORES]	44
FIGURA 3.33 CALIBRACIÓN DE LA BRÚJULA (COMPASS) [LOS AUTORES]	45
FIGURA 3.34 CONFIRMACIÓN DE CALIBRACIÓN ÉXITOSA DE LA BRÚJULA (COMPASS) [LOS AUTORES].....	46
FIGURA 3.35 CONFIRMACIÓN DE CALIBRACIÓN ÉXITOSA DEL GIROSCOPIO [LOS AUTORES].....	46
FIGURA 3.36 CALIBRACIÓN DEL ACCELERÓMETRO [LOS AUTORES]	47

FIGURA 3.37 NIVELACIÓN DEL HORIZONTE DEL SISTEMA AUTOPILOTO [LOS AUTORES]	48
FIGURA 3.38 PREPARACIÓN DEL RADIO CONTROL PARA SU CALIBRACIÓN [LOS AUTORES].....	48
FIGURA 3.39 INICIO DEL PROCESO DE CALIBRACIÓN [LOS AUTORES].....	49
FIGURA 3.40 SECUENCIA DE POSICIONES PARA CALIBRACIÓN DE MOVIMIENTO [LOS AUTORES]	49
FIGURA 3.41 CALIBRACIÓN DE PERILLAS Y SWITCHS [LOS AUTORES].....	49
FIGURA 3.42 ESCRITURA DE LOS NUEVOS PARÁMETROS EN LA MEMORIA DEL SISTEMA AUTOPILOTO [LOS AUTORES]	50
FIGURA 3.43 CONFIGURACIÓN DEL CANAL 5 EN EL CONTROL REMOTO [LOS AUTORES]	50
FIGURA 3.44 CONFIGURACIÓN DE LOS MODOS DE VUELO [LOS AUTORES]	51
FIGURA 3.45 AJUSTE DE PARÁMETROS DE LA BATERÍA [LOS AUTORES]	51
FIGURA 3.46 CONFIGURACIÓN DEL PARÁMETRO COM_POWER_COUNT [LOS AUTORES]	52
FIGURA 4.1 INSTALACIÓN Y CONFIGURACIÓN DEL LAUNCHER [LOS AUTORES]	54
FIGURA 4.2 CONEXIÓN ENTRE EL DISPOSITIVO ANDROID Y EL SERVIDOR [LOS AUTORES]	54
FIGURA 4.3 PERFIL DE MANDO PERSONALIZADO [LOS AUTORES].....	55
FIGURA 4.4 MENÚ DE INICIO DE PHOENIX R/C [LOS AUTORES]	55
FIGURA 4.5 PREPARACIONES PRELIMINARES [LOS AUTORES].....	56
FIGURA 4.6 DETECCIÓN Y CALIBRACIÓN DE LAS PALANCAS DEL MANDO [LOS AUTORES].....	56
FIGURA 4.7 DETECCIÓN Y CALIBRACIÓN DE LOS SWITCHS DEL MANDO [LOS AUTORES].....	57
FIGURA 4.8 CREACIÓN DEL PERFIL PARA EL MANDO [LOS AUTORES].....	57
FIGURA 4.9 SELECCIÓN DEL MANDO A CONFIGURAR [LOS AUTORES]	58
FIGURA 4.10 ASIGNACIÓN E INVERSIÓN DE CANALES [LOS AUTORES]	58
FIGURA 4.11 ELECCIÓN DE LA AERONAVE DJI PHANTOM [LOS AUTORES].....	59
FIGURA 4.12 ACTIVACIÓN DE LOS BINOCULARES [LOS AUTORES].....	60
FIGURA 4.13 SIMULACIÓN DE VUELO EN CAMPO ABIERTO [LOS AUTORES].....	61
FIGURA 4.14 SIMULACIÓN EN MEDIO DE LA CIUDAD [LOS AUTORES]	61
FIGURA 5.1 VENTANA DE VISTA DE VUELO DE QGROUNDCONTROL [LOS AUTORES].....	63
FIGURA 5.2 PESTAÑA DE VALORES Y PARÁMETROS DE MEDICIÓN [LOS AUTORES]	64
FIGURA 5.3 VISUALIZACIÓN DE LA ACTITUD Y LA BRÚJULA [LOS AUTORES].....	64
FIGURA 5.4 TESTEO DE LOS MOTORES [LOS AUTORES]	65
FIGURA 5.5 POSICIONAMIENTO Y ARMADO DE LA AERONAVE [LOS AUTORES]	65
FIGURA 5.6 PREPARACIÓN Y ENCENDIDO DE LA AERONAVE [LOS AUTORES]	66

LISTA DE TABLAS

TABLA 3.1 PINES DE CONEXIÓN DEL ESC	33
TABLA 6.1 COSTE DE EJECUCIÓN DE LAS ACTIVIDADES.....	68
TABLA 6.2 RECURSOS MATERIALES	68

CAPÍTULO 1

DEFINICIÓN DE LA PROPUESTA

1.1 INTRODUCCIÓN

El helicóptero fue considerada la primera aeronave de despegue y aterrizaje vertical, por la facilidad de este para ser utilizado en operaciones de rescate y corta duración, fue usado en muchas guerras y es usado hasta la actualidad con la limitación de su velocidad como la de los aviones. Con esta necesidad de tener rapidez en un aterrizaje y velocidad de vuelo nacen las aeronaves no tripuladas (UAV, Unmanned Aerial Vehicle o VANT vehículo aéreo no tripulado), o drones. A través del tiempo este tipo de naves fueron evolucionando, siendo ayuda para diferentes fines, en la industria, en el campo, en la seguridad, en la medicina entre otros. El uso de vehículos aéreos no tripulados ha crecido rápidamente a nivel comercial en los últimos años. Estos, se han convertido en elementos de trabajo, que ayudan a agilizar procesos y a llegar con informaciones más precisas y rápidas. Los vemos en muchos campos como en la fotografía aérea, como ayuda en el ejército y policía, como diversión en niños y adultos y se pueden utilizar en diferentes ámbitos para diferentes fines.

Según MuyCanal en su página web dice que “se estima que el uso de drones empresariales llegará hasta 46.000 envíos para el presente año, y, para el 2023 se triplicaría la cifra hasta aproximadamente 136.000 envíos”, este incremento de drones en todo el mundo reducirá en gran proporción mano de obra en diferentes actividades, reduciendo así el número de empleados, tanto en fotografía, en la policía, cuerpo de bomberos, ejército, salud, empresas de envío de mercancías, industrias, facilitando las tareas de forma más eficiente. [1]

Vemos que hay mucha demanda en la venta de VANT, pero muy poco conocimiento en nuestras universidades acerca de estas aeronaves, por ende, en estas instituciones hay mucha dificultad por falta de información. En esta investigación mejoraremos esta aeronave tipo VANT en su velocidad, en la calidad y economía del auto pilotaje, creando así una aeronave tipo VTOL (de sus siglas en inglés Vertical Take-Off and Landing ó despegue y aterrizaje verticales). En este trabajo mostraremos un autopiloto confiable, seguro y económico, llegando a una aeronave tipo VTOL, la cual se caracteriza por el despegue y aterrizaje vertical, así como el vuelo de forma horizontal con las características semejantes a las de otros aviones, con la diferencia que no se necesita pista para realizar despegues o aterrizajes. La ilustración que se muestra a continuación sirve para hacerse una idea del aspecto físico de una aeronave VTOL. En la Figura 0.1 se observa una aeronave VTOL de la marca Hélicéo, referencia Fusión:



Figura 0.1 Aeronave tipo VTOL Fusión [2]

1.2 PROBLEMA DE INVESTIGACIÓN

El uso de las aeronaves tipo VANT, va a incrementar rápidamente en los próximos años. Según la consultora de mercado Gartner, “quienes realizaron estimaciones sobre un posible crecimiento del 50% en el envío de drones empresariales impulsados por IOT (Internet of things, el internet de las cosas), hasta llegar a alcanzar un total de 526.000 unidades a nivel mundial, indicando también que dicho valor, para el año 2023, la cifra ascendería a 1.3 millones de unidades, llevando esto a la vez a un posible decremento en el número de empleados en diferentes industrias”. [1]

Un aumento en la demanda requiere de nuevos desarrollos tecnológicos, sin embargo muchas de las aeronaves VANT ofrecidas en el mercado son inestables, generando ciertos accidentes, por ende se requieren aeronaves desarrolladas de forma confiable, seguras y económicas, además de ello, este sector es bastante limitado, lo que conlleva a que no se pueda generar una mayor oferta en diferentes sistemas autopilotos incluso a nivel investigativo y de academia para las universidades, por otra parte estos sistemas son costosos, complejos de construir y de diseñar desde cero, además de esto, hay muy poca información de su programación, estructura y control. Por otro lado, se tiene que los autopilotos existentes actualmente vienen configurados de forma preestablecida, teniéndose así que el “drone” viene listo para volar, sin embargo, la persona, grupo de investigación o empresa que quiera investigar alrededor de este tema, desconoce cómo funciona y no pueden innovar o desarrollar nuevos sistemas o prototipos.

Pasando a otra categoría de aeronaves se encuentran las de tipo VTOL, las cuales, son más efectivas en cuanto a velocidad de desplazamiento y cuya constitución amplía el rango de posibles aplicaciones respecto a las aeronaves tipo VANT, sin embargo, las VTOL no se encuentran fácilmente en el mercado lo que conlleva a que en la mayoría de los casos se deba hacer un proceso de desarrollo y construcción de esta de forma particular.

¿Es posible desarrollar un sistema programable micro controlado para el control de una aeronave de despegue vertical y vuelo horizontal?

1.3 DELIMITACIÓN

La idea del proyecto es desarrollar un sistema programable micro controlado para el control de una aeronave de despegue vertical y vuelo horizontal usando tecnología disponible en el mercado actual y diseñando un prototipo de fabricación sencilla, diseñando así un control de orientación que logre estabilizar en la posición de coordenadas (0, 0, 0) a un vehículo aéreo tipo VTOL. En otras

palabras, se trata de encontrar una ley de control que, una vez estabilizada la orientación del vehículo, sea capaz de estabilizar la posición de la aeronave al origen del marco de referencia y adicionalmente pueda mantenerlo en esa posición aún bajo la acción de perturbaciones del medio ambiente.

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema programable micro controlado para el control de una aeronave de despegue vertical y vuelo horizontal.

1.4.2 Objetivos específicos

- Identificar los diferentes sistemas autopilotos ya existentes en el mercado y sus características de funcionamiento básicas.
- Programar un sistema autopiloto mediante componentes de hardware y software.
- Configurar sistemas software y hardware de un autopiloto.
- Realizar simulaciones mediante software especializado.
- Validar funcionamiento en una plataforma aérea de vuelo vertical.

1.5 JUSTIFICACIÓN

Con el paso de los años, el mundo de la aeronáutica ha venido evolucionando de manera constante, llevando a cabo desarrollos en los diferentes ámbitos, como lo es la implementación de nuevos materiales, la construcción de gran variedad de aeronaves, así como también se ha logrado una modernización de las mismas, implementando sistemas de control automatizados que permiten que dichas aeronaves, mediante la configuración de ciertos parámetros, se puedan manejar de forma autónoma bajo ciertas condiciones de vuelo. Existen tanto diferentes marcas como modelos de sistemas de piloto automático los cuales según la configuración asignada se encargan de controlar las superficies de control las cuales varían según el tipo de aeronave en donde se va a implementar, ya sea un aerodino (de ala fija o de ala giratoria), un dirigible, un vehículo aéreo no tripulado (VANT), entre otras, sin embargo, al momento de diseñar un nuevo tipo de aeronave, se requiere por ende de un piloto automático que se pueda adaptar a las necesidades de la misma, y si esta se sale de las configuraciones comunes de aeronaves, es necesario proceder a diseñar y configurar un nuevo sistema de autopiloto, convirtiéndose esto en una labor de alto grado de dificultad, ya que se carece de la información necesaria en los medios para llevar a cabo este objetivo.

Para el presente proyecto, se propone entonces un sistema programable micro controlado para el control de aeronaves tipo VTOL, tomando como base un sistema autopiloto Pixhawk, el cual, con su programación y configuración respectiva (integración de hardware y software), nos da la posibilidad de contar con un autopiloto que puede ser implementado en nuevos tipos y diseños de aeronaves, además, cuyo desarrollo requiere entrar en temas como lo es diseño, control y mecánica. El sistema autopiloto será integrado en una aeronave no

tripulada de vuelo vertical y horizontal, la cual, será desarrollada en el laboratorio de aeronáutica de la Universidad Tecnológica de Pereira, y será diseñada de modo que los gastos de construcción mecánico sean asequibles, posteriormente el ensamble mecánico y eléctrico logran una máquina que satisfaga en parte las necesidades del ambiente, lo que conlleva a su vez a que el consumo de energía sea óptimo para el buen funcionamiento de la misma.

CAPÍTULO 2

VEHÍCULOS AÉREOS NO TRIPULADOS

Los vehículos aéreos no tripulados o llamados (VANT, o UAV de sus siglas en inglés: unmanned aerial vehicle) conocidos como drones, son aeronaves que no necesitan ser tripuladas para su funcionamiento, se caracterizan por su despegue y su aterrizaje vertical, con un movimiento horizontal lento, sin forma de planear para un aterrizaje en caso de emergencia. Son robots pequeños y ligeros, realizados en fibra de carbono, materiales metálicos y un poco de plástico, son manejados remotamente a través de un control manual fácil de manejar, puede ser controles de smartphone, o radio, por medio de los cuales se aumenta o disminuye la aceleración y la altura. Los más comunes tienen 4 motores ubicados en sus 4 ejes, estos motores son pequeños y redondos, los cuales tienen el mismo principio de vuelo que los helicópteros para mantenerse en el aire. La capacidad de transmisión de datos depende de la cantidad de energía que le queda disponible en baterías para navegar.

Una característica importante de los VANT es que en algunas ocasiones se les incorpora una cámara de vídeo, a la cual se le incorporan transmisores y receptores a determinada frecuencia de trabajo, que permite transmitir el vídeo a un dispositivo remoto a distancias largas, además dichas cámaras suelen encontrarse por un precio asequible.

Actualmente los drones han tenido un gran avance, siendo usados para diferentes actividades que van desde uso lúdico hasta investigaciones científicas, rescates, vigilancia fronteriza, uso de grabaciones comerciales, etc.

2.1. CLASIFICACIÓN

Los VANT se clasifican dependiendo de su misión principal de la siguiente manera: [3]

- Blanco: Son de utilidad para llevar a cabo simulaciones de aviones o ataques a enemigos en los sistemas de defensa, ya sea tierra o aire.
- Reconocimiento: Se utilizan para la obtención y envío de información militar. Entre ellos, los principales son los MUAV (micro unmanned aerial vehicle) de tipo avión o helicóptero.
- Combate (UCAV, unmanned combat air vehicle): Son utilizados para combatir y para el desarrollo de misiones que impliquen alto riesgo.
- Logística: Empleados para transporte de mercancía.
- Investigación y desarrollo: Son objeto de prueba en investigaciones de sistemas que se encuentran en desarrollo.

- UAV comerciales y civiles: Son aquellos diseñados para propósitos civiles como lo son la toma de fotografías y grabación de vídeo en alta calidad.

En la Figura 0.1, Figura 0.2 y Figura 0.3 se muestran algunos ejemplos de UAVS:



Figura 0.1 UAV de reconocimiento Korsar de origen ruso [4]



Figura 0.2 UAV de combate General Atomics MQ-9 Reaper [5]



Figura 0.3 DJI Phantom 4 Pro V2.0 [6]

2.2. AERONAVES DE DESPEGUE VERTICAL Y VUELO ORIZONTAL (VTOL)

La principal característica en este tipo de aeronaves es el despegue y aterrizaje verticales, así como el vuelo de forma horizontal rápido, con características de vuelo semejantes a las de otros aviones que, en contraparte, para poder despegar, requieren de una pista para poder tomar velocidad y poder alzar vuelo. Para conseguir despegar o aterrizar de manera vertical estas aeronaves se caracterizan por utilizar motores a reacción giratoria que en algunos tipos de aeronaves estos al comienzo del despegue se ponen de forma vertical y después

poco a poco rotan hasta ponerse de forma horizontal una vez hayan adquirido la velocidad necesaria para volar. Por otro lado, una variante también utilizada es la de aeronaves que emplean un sistema multi rotor como los QuadCopter, con la diferencia de que estas cuentan también con alas fijas como los aviones, así como con una hélice en la parte frontal, donde esta última es quien permite su desplazamiento horizontalmente tras haber ascendido a determinada altura de forma vertical con el sistema multi rotor. Una aeronave tipo “vertical Take-off and Landing” de las siglas VTOL, es un prototipo muy útil a la hora de entender el fenómeno aerodinámico que se produce en las máquinas voladoras que pueden desplazarse y mantenerse suspendidas en el aire.

2.3. SISTEMA AUTOPILOTO

Es un sistema de control usado para guiar y controlar aeronaves sin la ayuda del ser humano. Este término alude especialmente a aviones, aunque también a barcos o automóviles. En el mercado actual se pueden encontrar diferentes sistemas autopilotos cuya configuración y características varían acorde al tipo de vehículo o aeronave para el cual están diseñados. Los autopilotos o pilotos automáticos son cada vez más comunes, están haciendo presencia en diferentes ambientes donde se necesite hacer recorridos establecidos, por ejemplo, hacer un recorrido de fumigación en un campo abierto, donde debe recorrer diferentes puntos.

2.3.1. Clasificación

2.3.1.1. Básicos

Se caracterizan por ser supremamente fáciles de programar, instalar y configurar. Entre las más comunes se encuentran: iNav, OpenPilot, CC3D:

- **iNav:** Es esencialmente una bifurcación de BetaFlight que se enfoca en agregar funciones de drones autónomos en lugar de funciones de carreras FPV (de las siglas First-person View ó vista en primera persona). Esto permite que el software vuele su dron a puntos de referencia o regrese a casa por su cuenta. Este firmware es compatible con la mayoría de los controladores de vuelo de carreras FPV. iNav es mucho más barato a comparación de las placas ArduPilot, también es compatible con aviones multi rotores y aviones de ala fija. En la Figura 0.4 se muestra la placa y el software de control respectivo:



Figura 0.4 Sistema autopiloto iNavy y software de control [7]

- **NAZE 32:** Es una placa de control basada en el procesador STM32 corriendo a 72MHz desarrollada por el usuario timecop de rcgroups.com. Son muy buenas, sencillas fáciles de programar y ensamblar, pero son costosas. Es una de las más usadas para realizar proyectos caseros, adicional a que también es usada en micro multicopters para realizar acrobacias y desarrollar velocidad. La última versión viene con nuevas mejoras en cuanto a distribución de pads para realizar las soldaduras y mejoras en cuanto a sensores. En la Figura 0.5 se muestra la placa de control Naze32. [8]



Figura 0.5 Placa de control Naze32 [8]

- **OpenPilot:** La controladora de vuelo OpenPilot CC3D fue desarrollada por la comunidad de OpenPilot y desde que salió al mercado ha sido usada en infinidad de proyectos, demostrando su flexibilidad al poder usarse en tricópteros, quads fpv, hexas, octo y alas volantes. Proporciona un vuelo muy suave y estable gracias a su controlador STM de 32bits. En la Figura 0.6 se muestra la placa de control OpenPilot CC3D EVO. [9]



Figura 0.6 Placa de control Openpilot CC3D EVO [9]

- **LibrePilot:** Esta tarjeta de control fue creada para aplicaciones que implican el control y estabilización de vehículos aéreos no tripulados y robots. Este proyecto obtuvo sus bases desde OpenPilot y se centra en ser un sistema autopiloto de vuelo general que admita aviones de ala fija y rotores. Esta cuenta a su vez con una interfaz gráfica de usuario bastante agradable, así como con un firmwares estable y sólido para diferentes trabajos. Adicional a ello, para sus capacidades es lo suficientemente potente y económico, sin embargo, no cuenta con tantas funciones a comparación de ArduPilot, lo que lo vuelve un sistema autopiloto poco popular actualmente. [10]

A continuación, en la Figura 0.7 se muestra una imagen de la tarjeta de control LibrePilot y una captura de la interfaz del software de manejo:



Figura 0.7 Tarjeta de control LibrePilot y software de manejo [11]

2.3.1.2. De fin específico

Se utilizan en aeronaves cuya función es puntual, ejemplo de ellas como lo son de: espionaje, fotografía, fines nucleares, entre otras. A continuación, se describen algunas de las tarjetas de control de fin específico más conocidas:

- **NAZA DJI:** NAZA es llamado a un sistema de control de vuelo usado por lo general para fotografía, espionaje, trabajos nucleares entre otros

trabajos de precisión. Su fabricante es la empresa DJI, la cual es ampliamente reconocida a nivel mundial por ofrecer al mercado vehículos aéreos no tripulados profesionales para su uso en fotografía y videografía, así como otros periféricos relacionados, llegando a representar más del 70% en el mercado de drones hablando a nivel mundial. Su tecnología de drones con cámara se ha utilizado ampliamente en las industrias, la música, la televisión y el cine. Los productos de la empresa también han sido utilizados por fuerzas militares y policiales, así como por grupos terroristas varias agencias a nivel local y federal continúan utilizando productos DJI. En la Figura 0.8 se muestra la placa de control NAZA DJI: [12]



Figura 0.8 Placa de control NAZA DJI [12]

- **ArduPilot:** Este sistema de piloto automático puede ser considerado como el más popular en el momento, ya que lleva un largo recorrido a lo largo del tiempo y tienen una comunidad de desarrollo activa ya que es de código abierto. El software de ArduPilot está dividido en tres ramas, y puede ser utilizado para controlar gran variedad de vehículos, entre ellos: Cuadricópteros, aviones de ala fija, barcos, rovers e híbridos.

ArduPilot se caracteriza por brindar la posibilidad de crear y usar sistemas de vehículos aéreos no tripulados de tipo autónomo y con altos índices de confiabilidad. Entre algunas de sus aplicaciones se encuentra que puede ser utilizado para el desarrollo de misiones de vuelo autónomo en diferentes puntos previamente establecidos. En la Figura 0.9 se muestra la placa de control ArduPilot: [13]



Figura 0.9 Placa de Control ArduPilot [14]

- **PIXHAWK PX4:** Es otro piloto automático potente y con todas las funciones propias de un autopiloto. Este admite múltiples tipos de vehículos, también tiene una comunidad de desarrolladores activa. En la superficie, puede considerarse que esta placa es similar a ArduPilot, ya que hay una gran cantidad de hardware que puede ejecutar firmware PX4 o ArduPilot, los cuales son software de control de vuelo de código abierto para drones y otros vehículos no tripulados), ambos sistemas utilizan el mismo protocolo de telemetría (MAVLink: Micro Air Vehicle Link, que es un protocolo para comunicarse con vehículos aéreos no tripulados), además una gran cantidad de softwares son compatibles con ambos, como QGroundControl. La diferencia es cuando se necesita hacer funciones muy específicas puede notar algunas variaciones entre los dos proyectos y la licencia del software, PX4 usa BSD (sistema operativo derivado de Unix) mientras que ArduPilot usa GPL (Licencia Pública General de GNU o más o por sus siglas en inglés General Public License). Esto hace que PX4 sea más atractivo para uso comercial, ya que no es necesario que sus modificaciones sean de código abierto. Con ArduPilot, cualquier cambio debe compartirse como código abierto. Entonces, si planea desarrollar alguna función especial que desea mantener en privado o vender, PX4 es el camino para seguir.

2.3.1.3. De alto desempeño

Están diseñadas para acrobacias y alta velocidad. Entre las principales se encuentran:

- **BetaFlight:** Este sistema es utilizado principalmente para cuadricópteros, y tiene soporte para aviones FPV de ala fija. Los desarrolladores de esta plataforma se centran en obtener la mayor velocidad y rendimiento de su controlador de vuelo y QuadCopter, con cosas como leer datos de sensores a 32 kHz y enviar actualizaciones al motor a la misma velocidad increíblemente rápida. Admite una gran cantidad de controladores de vuelo y tiene una lista impresionante de compatibilidad con funciones avanzadas de FPV como control OSD (por sus siglas en inglés On Screen Display que significa visualización en pantalla. En la Figura 0.10 se muestra la placa de control BetaFlight y una captura de su respectivo software de control: [15]



Figura 0.10 Placa de control y software de BetaFlight [15]

- **CleanFlight:** CleanFlight se creó a partir de BaseFlight, con el fin de mantener el código y facilitar el trabajo. Este rápidamente ganó popularidad y fue el firmware más popular hasta que apareció BetaFlight. Recientemente, CleanFlight se fusionó nuevamente con BetaFlight (desde v2.0) para volver a actualizarlo con todas las características que los destacan. A continuación, en la Figura 0.11, se muestra la placa de control CleanFlight:

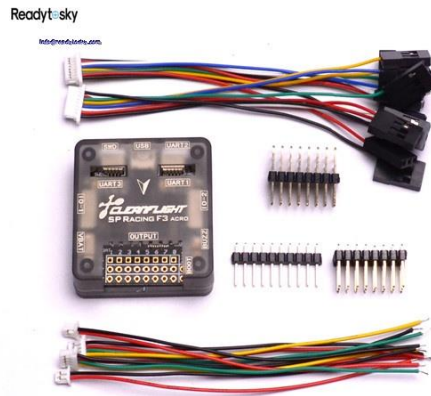


Figura 0.11 Placa de control CleanFlightv [16]

- **DRonin:** OpenPilot tiene varias ramificaciones, DRonin es otra, esta tiene un enfoque principal en las carreras FPV utilizando el firmware de OpenPilot. La característica más importante de DRonin es que tiene un programa de autoajuste, quiere decir que se ajusta automáticamente a la configuración de su controlador de vuelo para su Quad dando la mejor respuesta en su QuadCopter. Sin embargo, muchas de las otras características específicas de FPV vienen con BetaFlight. Este es un firmware de control de vuelo sofisticado, con excelentes asistentes de configuración en hardware asequible basado en CC3D. [17]
- **KISS:** Siglas en inglés que significan: Keep It Super Simple (KISS), es el único proyecto de código cerrado, KISS solo se ejecuta en tableros KISS de Flyduino. KISS ha ganado mucha popularidad por tener un hardware confiable, es fácil de usar los desarrolladores de KISS introdujeron nuevas características obligando a los proyectos de código abierto a mejorar sus prototipos que obligaron a los proyectos de código abierto a ponerse al día durante mucho tiempo. En la Figura 0.12 se muestra la placa de control KISS:



Figura 0.12 Placa de control KISS

- **RaceFlight:** Este es una ramificación de CleanFlight con un enfoque en las carreras FPV y los procesadores F4, RaceFlight sea conocido por su suavidad y alta respuesta. Posee un software muy bueno para configurar su controlador de vuelo. En su momento, CleanFlight solo admitía F3. Sin embargo, después de un tiempo, los desarrolladores comenzaron a mover el código de RaceFlight para eventualmente hacer que todo el proyecto fuera de código cerrado y solo funcionara en los tableros oficiales de RaceFlight.

2.3.1.4. Obsoletos

Como es de esperarse con el paso del tiempo hay nuevos desarrollos tecnológicos, por lo que algunos sistemas autopilotos pasan a ser obsoletos y por ende poco utilizables dada su baja capacidad de procesamiento, velocidad de respuesta, entre otros factores. Entre algunos de ellos están:

- **MultiWii 328p:** Este comenzó casi al mismo tiempo que se lanzó la Nintendo Wii, fue la génesis del firmware moderno de QuadCopter. Wii les dio a todos acceso a sensores IMU económicos. Este proyecto combinó un MCU Arduino de 8 bits y un Wiimote. Aunque no está en desarrollo activo, merece mencionarlo en este proyecto de grado. En la Figura 0.13, se muestra la placa de control MultiWii:



Figura 0.13 Placa de control MultiWii [18]

- **BaseFlight (Vuelo base):** Este fue el comienzo para muchos otros firmwares FPV de 32 bits. Este es creado antes que BetaFlight, ya que fue el primer controlador de vuelo FPV de 32 bits. BaseFlight fue una reescritura completa

del firmware MultiWii adaptado para usar hardware de 32 bits reemplazando al de 8 bit.

- **TauLabs:** Es una derivación de OpenPilot, dirigida a profesionales. Su fin fue crear código de primera calidad, adicional a esto que fuera reutilizable y facilitar el uso para la investigación. Al igual que otras derivaciones de OpenPilot, En el momento TauLabs tiene muy poco soporte de hardware, aunque el código todavía está disponible para quien quiera tener un código de alta calidad. En la Figura 0.14 se muestra la placa de control TauLabs:

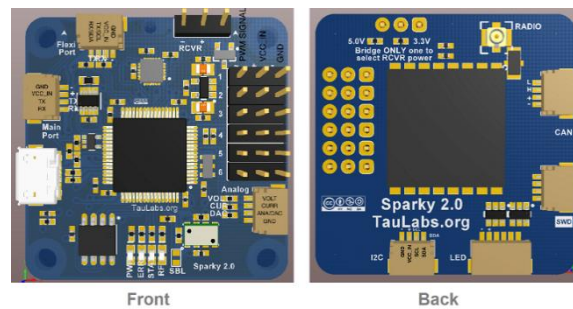


Figura 0.14 Placa de control TauLabs [19]

2.4. PIXHAWK 2.4.8

Esta placa de control cuenta con diferentes prestaciones para conformar diferentes tipos tanto de vehículos como aeronaves, a continuación, se explica en detalle cada una de ellas, así como las conexiones físicas de la tarjeta en la Figura 0.15: [20]



Figura 0.15 Placa de control Pixhawk 2.4.8 e indicadores de puertos de conexión

2.4.1. Características principales

- **MICROPROCESADOR:** La placa es gobernada por un procesador ARM Cortex M4, el cual se caracteriza por ser de alto rendimiento y cuya arquitectura es de 32 bits, teniendo este la posibilidad de ejecutar sistemas operativos en tiempo real como NuttX RTOS. Entrando más en detalle, dicho procesador cuenta con 2 MB de memoria Flash, 256KB de memoria RAM y una frecuencia de reloj de 168 MHz, así como con un procesador de respaldo STM32F103 cuya arquitectura es también de 32 bits.

- **SENSORES:** El autopiloto cuenta con un giroscopio digital de 3 ejes referencia L3GD20, es utilizado para la medición de tasas angulares de rotación. Además de ello viene configurado con 2 acelerómetros y magnetómetros: LSM303D y MPU6000, los cuales son de 3 y 6 ejes respectivamente. Finalmente se tiene que esta tarjeta de control cuenta con un barómetro MS5607 de alta precisión.

- **CONEXIONES:**

- ☐ UART (por sus siglas en inglés Universal Asynchronous Receiver-Transmitter o Transmisor-Receptor Asíncrono Universal) compatible con alto voltaje y con control de flujo de hardware.
- ☐ 2 puertos CAN (de sus siglas en inglés *Controller Area Network*).
- ☐ Entrada Spektrum compatible con receptor de satélite.
- ☐ Entradas y salidas compatibles con SBUS (sistema de bus de computadora) de Futaba.
- ☐ Entrada de señal PPM.
- ☐ Entrada RSSI (PWM o Voltaje).
- ☐ I2C.
- ☐ SPI.
- ☐ Entrada de 3,3 V y 6,6 V CC.
- ☐ Interfaz micro USB externa.
- ☐ 14 salidas de PWM/servo (8 principales y 6 auxiliares).

- **DIMENSIONES Y PESO:**

- ☐ Peso 38 g (1,3 oz)
- ☐ Ancho 50 mm (2,0 ").
- ☐ Altura 15,5 mm (0,6 ").
- ☐ Longitud 81,5 mm (3,2 ").

- **OTRAS FUNCIONES:** La Pixhawk 2.4.8 consta de un sistema de alimentación redundante y un control de respaldo a fin de incrementar su seguridad, fuera de ello registra los datos de vuelo en una tarjeta microSD. Por otro lado, tiene modo de vuelo tanto automático como manual.

- **CONTENIDO DEL KIT ADQUIRIDO:** Adicional a la tarjeta de control, el sistema autopiloto cuenta además con una serie de componentes que se usan ya sea como elementos de seguridad, para configurar en sí la aeronave o para habilitar funciones adicionales en el sistema.

Se tiene entonces por un lado un botón de seguridad que permite accionar la aeronave de forma manual cuando ya se esté en condiciones

adecuadas para el vuelo, un buzzer (timbre), el cual mediante secuencias de sonido nos indica el estado del autopiloto y con las propias luces que trae la tarjeta para dicho propósito.

Para poder ubicar el sistema autopiloto en un sistema de coordenadas terrestres, el kit consta de un Módulo GPS Neo-M8N, el cual a su vez tiene su propia brújula que se comunica con la placa por comunicación i2c, este viene también con su respectivo soporte para fijarlo a la estructura principal de la aeronave.

Por otra parte, se cuenta también con diferentes módulos, como lo son

- ☐ RGB, el cual sigue la misma secuencia de luces de la tarjeta principal y que cuenta con un puerto USB, lo que facilita la conexión para su programación tras el ensamble de la aeronave.
- ☐ PPM, el cual permite realizar la codificación de hasta 8 señales de PWM (modulación por ancho de pulso) en una sola señal PPM (modulación por posición de pulso).
- ☐ Divisor de ampliación i2c, que se utiliza para poder conectar a la tarjeta diferentes periféricos y sensores que manejan dicho tipo de comunicación.
- ☐ Minim OSD, que es una placa de circuito que extrae diferentes datos de telemetría del sistema autopiloto y que los transmite a un monitor para su visualización en primera persona.
- ☐ Alimentación (con conector XT60), el cual regula el voltaje que va de la batería a la placa y a controladores de velocidad de los motores.
- ☐ Telemetría, con frecuencia de trabajo de 433 MHz para este caso.

El firmware, por otro lado, es almacenado en una tarjeta SD de 16gb que es insertada en su ranura correspondiente de la tarjeta de control.

Finalmente se tiene una serie de cables adaptadores con diferentes tipos de clavijas para llevar a cabo conexiones que así lo requieran.

- ☐ Cable adaptador de 6 clavijas.
- ☐ Cable adaptador de 4 clavijas.
- ☐ Cable de 3 clavijas de DuPont.

2.5. SOFTWARE OPEN SOURCE

En español Open Source significa de código abierto, es decir, que está diseñado para que su código fuente sea accesible al público, de forma que cualquiera pueda verlo, modificarlo, y distribuirlo a su parecer. Regularmente se desarrolla

de forma colaborativa y descentralizada, donde se depende de la comunidad para la detección de errores y su solución. Este tipo de software es flexible, duradero y mucho más económico si se le compara con los desarrollos de otras empresas, por lo que se ha convertido en una forma de trabajo debido a que sirve para explorar y encontrar nuevas formas de resolver problemas

2.5.1. Software de programación y de control

2.5.1.1. APM PLANNER 2.0

APM Planner 2.0 es una aplicación de estación terrestre de código abierto para pilotos automáticos basados en MAVLink, que es un protocolo de mensajería muy ligero para comunicarse con drones. incluidos APM y PX4 / Pixhawk, que se pueden ejecutar en Windows, Mac OSX y Linux. APM Planner 2.0 está muy influenciado por la sencilla interfaz de usuario de Mission Planner y la capacidad multiplataforma de QGroundControl.

APM Planner 2.0 es el hijo de Mission Planner y QGroundControl, que combina la sencilla interfaz de usuario de Mission Planner y la capacidad multiplataforma de QGroundControl. En la Figura 0.16 se muestra una captura de la interfaz del programa APM Planner 2.0: [21]

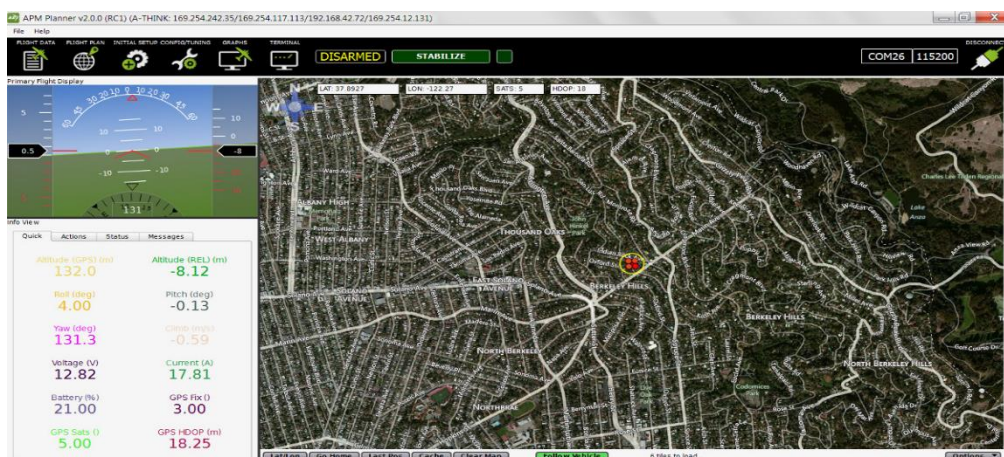


Figura 0.16 Interfaz del programa APM PLANNER 2.0 [21]

2.5.1.2. QGroundControl

Es una aplicación para configurar y volar un piloto automático basado en PX4. Es multiplataforma y es compatible con todos los principales sistemas operativos: Windows, Linux, Mac OS adicional a Móviles compatibles con Android e iOS (actualmente enfocado en tabletas), este proporciona un control de vuelo completo además de configuración del vehículo para vehículos con motor PX4 o ArduPilot. Uso fácil y directo para principiantes, y ofrece soporte de funciones de alta gama para usuarios experimentados.

CARACTERÍSTICAS PRINCIPALES: QGroundControl posee una configuración completa de vehículos con ArduPilot y PX4. Adicionalmente brinda soporte de vuelo para vehículos que ejecutan cualquier tipo de piloto automático cuyo protocolo de comunicación sea MAVLink. Este software brinda la posibilidad de planificar misiones para vuelo automático,

visualizando el mapa de vuelo mostrando: Posición, puntos de referencia y la ruta de vuelo; así como también permite la transmisión de video con superposiciones de pantalla. Cabe aclarar que también es posible diligenciar múltiples vehículos.

En la Figura 0.17 se muestra una captura de la interfaz del programa QGroundControl: [22]

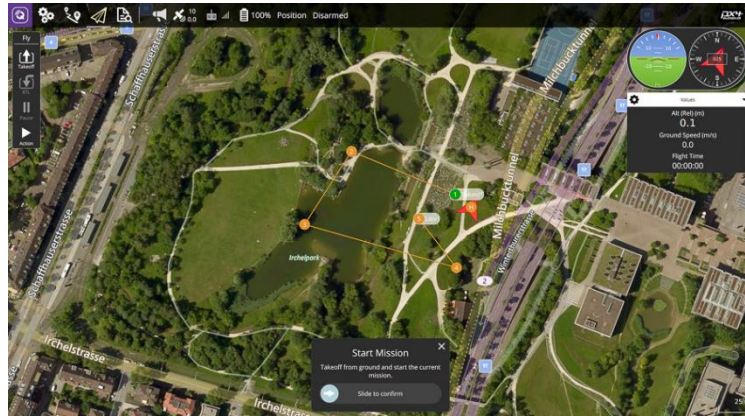


Figura 0.17 Interfaz del programa QGroundControl [22]

2.5.1.3. Mission Planner

Mission Planner es una estación de control de tierra para APM: Plano, APM: Helicóptero y APM: Rover. Mission Planner se puede utilizar como una utilidad de configuración o como complemento de control dinámico de un vehículo autónomo.

CARACTERÍSTICAS PRINCIPALES:

- Entrada de punto de ruta / valla / punto de reunión con apuntar y hacer clic, utilizando Google Maps / Bing / Mapas de calles abiertos
- Seleccionar comandos de misión de los menús desplegables
- Descargue los archivos de registro de la misión y analícelos
- Configure los ajustes del piloto automático para su vehículo
- Interfaz con un simulador de vuelo de PC para crear un simulador UAV completo de software en el bucle (SITL: Software in the loop)).
- Ejecute su propia simulación SITL de muchos tipos de marcos para todos los vehículos ArduPilot.
- Utilice los menús de la barra lateral para obtener instrucciones y más información.

En la Figura 0.18 se muestra una captura de la interfaz del programa Mission Planner: [23]



Figura 0.18 Interfaz del programa Mission Planner [23]

2.5.2. Software de simulación

Mediante el uso de simuladores es posible utilizar el código de vuelo PX4 para el control de un vehículo modelado por computadora en un entorno simulado, de esta forma pudiendo así interactuar con el mismo como se haría con un vehículo real, utilizando ya sea QGroundControl, una API externa o un radiocontrol/Gamepad.

PX4 soporta 2 tipos de simulaciones: Software in the loop (SITL) y Hardware in the loop (HITL), donde para el primero, la simulación se ejecuta directamente en una computadora y para el segundo, usando un firmware de simulación en la tarjeta de control física, para este caso la PIXHAWK 2.4.8.

A Continuación, se brinda información sobre algunos de los simuladores más relevantes:

2.5.2.1. SIMULADORES SITL (SOFTWARE IN THE LOOP)

- **jMAVSim:** Se caracteriza por ser un simulador de vuelo de aeronaves multi rotor/Quad simple, ya que permite volar vehículos tipo helicóptero con PX4 en un entorno completamente simulado. Este software es fácil de configurar, y entre algunas de sus pruebas se hallan las de despegue, vuelo, aterrizaje y respuesta adecuada en diferentes condiciones donde se presenten falencias. Por otra parte, este software está disponible para los 3 sistemas operativos existentes como lo son MacOS, Windows y Linux. En la Figura 0.19 se muestra una captura del entorno de simulación de jMAVSim: [24]

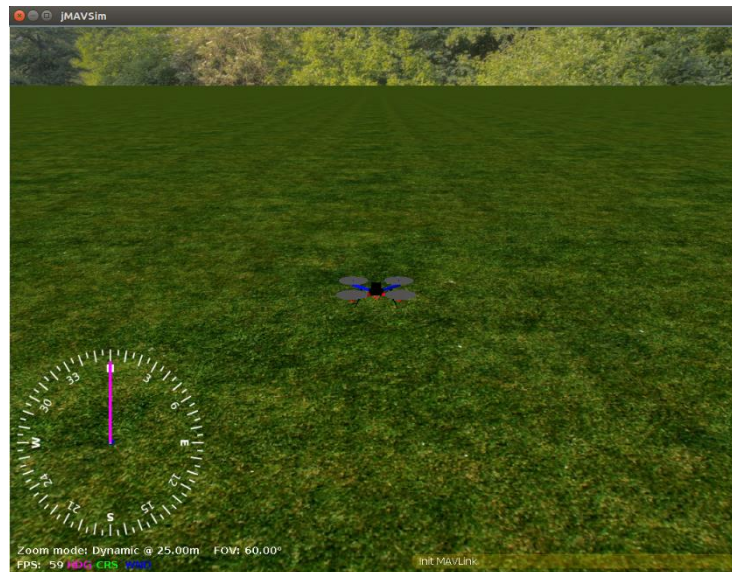


Figura 0.19 Visualización del entorno de simulación de jMAVSim [24]

- **FlightGear:** Es un simulador diseñado para simulaciones de aeronaves en diferentes condiciones climáticas. Este programa se caracteriza por ser altamente competitivo con simuladores de vuelo comerciales ya que es de código libre. La Figura 0.20 muestra una captura del entorno de simulación de FlightGear: [25]

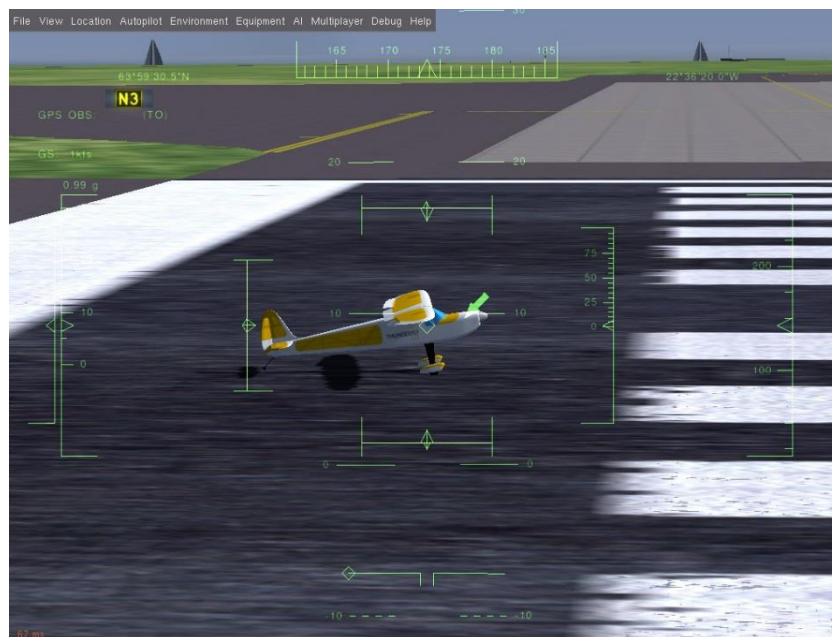


Figura 0.20 Visualización del entorno de simulación de FlightGear [25]

2.5.2.2. SIMULADORES HITL (HARDWARE IN THE LOOP)

- **GAZEBO:** Es un entorno de simulación 3D de alta capacidad diseñado para robots autónomos. Este a su vez es adecuado para probar la evasión de objetos y la visualización por computadora. Entre algunos de los vehículos que pueden ser simulados se hallan: Submarinos, barcos, vehículos terrestres, aeroplanos y diferentes configuraciones de aeronaves multi rotor.

En la Figura 0.21 se muestra un vehículo terrestre situado en el entorno de simulación de Gazebo: [26]

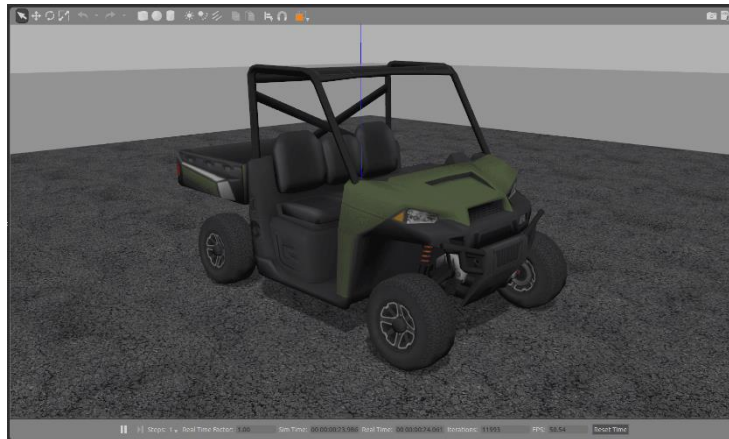


Figura 0.21 Ejemplo de vehículo terrestre simulado en Gazebo [26]

A diferencia de jMAVSim, Gazebo solo está disponible para MacOS y Linux y por otro lado también se caracteriza por permitir llevar a cabo simulaciones de tipo HITL en diferentes tipos de aeronaves como lo son las multi rotor y las de tipo VTOL. En la Figura 0.22 se muestra una aeronave tipo VTOL situada en entorno de simulación de Gazebo:

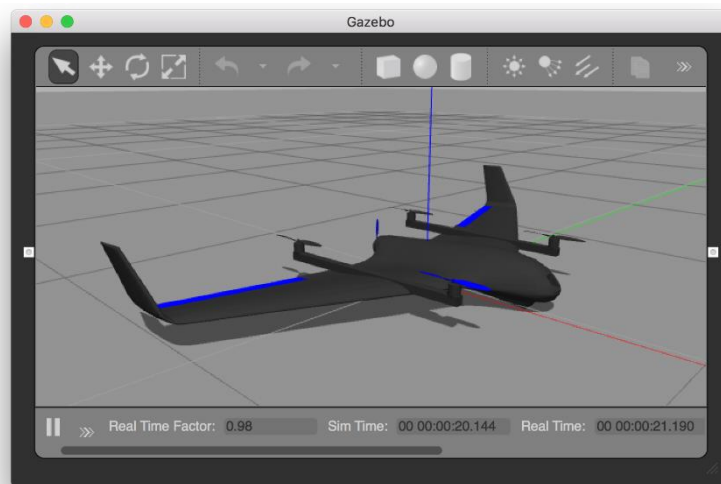


Figura 0.22 Ejemplo de VTOL simulado en Gazebo [26]

- **AirSim:** Es un simulador de código abierto construido sobre Unreal Engine y diseñado para drones, automóviles, entre otros. Se caracteriza por ser multiplataforma y por admitir hardware en bucle con la mayoría de los controladores de vuelo como lo es PX4, permitiendo llevar a cabo simulaciones tanto física como visualmente. El objetivo de este simulador es ofrecerse como una plataforma de inteligencia artificial para experimentar con algoritmos de aprendizaje, visión por computadora y otras aplicaciones para vehículos autónomos. Se propone también una API empleada para la recuperación de datos, así como para el control de vehículos de forma independiente de la plataforma. En la Figura 0.23 se muestra un ejemplo de los entornos de simulación disponibles en AirSim: [27]



Figura 0.23 Ejemplo de entorno de simulación en AirSim [27]

2.5.2.3. Simulador Phoenix R/C 5.5

Se caracteriza por ser un simulador de vuelo de alta calidad, el cual se destaca por ser compatible con gran variedad de emisores (radio controles) disponibles en el mercado, mediante el uso de su adaptador respectivo. Este simulador cuenta con variedad de entornos de vuelo, efectos gráficos y aeronaves como helicópteros, aviones de ala fija, multi rotores, etc. Se encuentra también que cada uno de los componentes de los modelos suministrados por Phoenix es modificable. Por otra parte, cuenta con un modo de competición, que se puede utilizar para mejorar las técnicas de vuelo. En la Figura 0.24 se muestra la pantalla de carga y el menú de inicio del programa: [28]

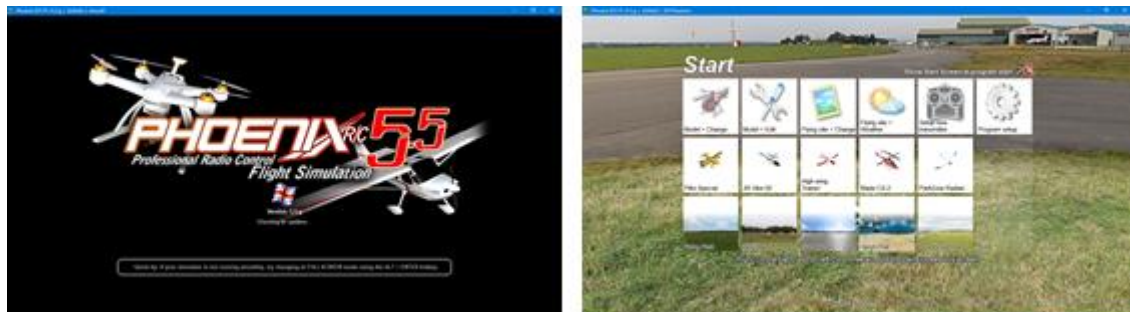


Figura 0.24 Pantalla de carga y menú de inicio del programa [Los autores]

2.5.2.4. DroidJoy

Es una aplicación móvil que permite usar dispositivos Android como mando de control personalizable de una computadora. Para su uso, es necesario descargar también una aplicación que en el ordenador que actúa como servidor y que se comunica vía Wifi. La pantalla de carga de la aplicación y el menú principal del servidor se muestran en la Figura 0.25:[29]

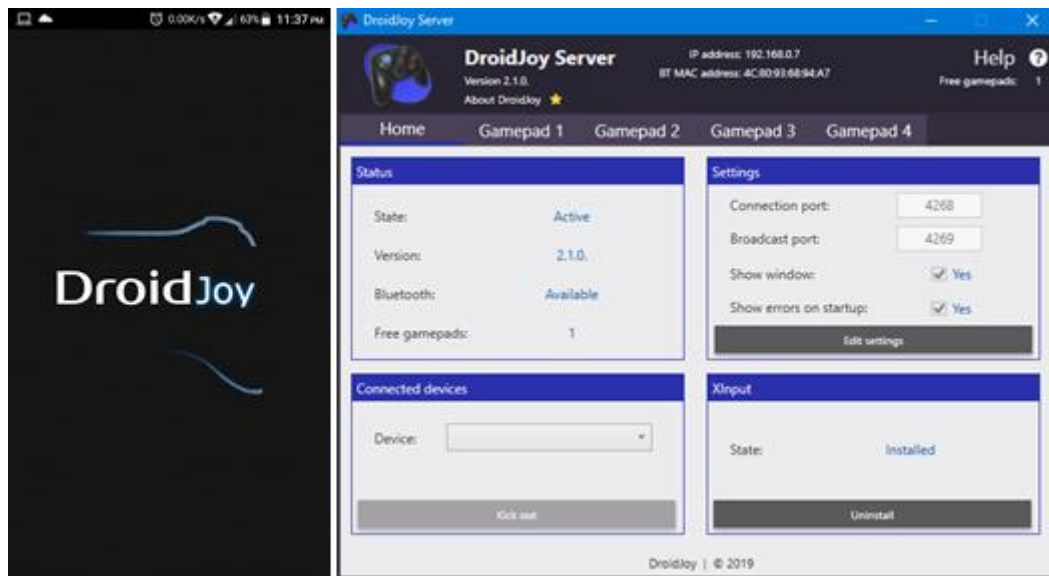


Figura 0.25 Pantalla de carga de DroidJoy app y menú principal del servidor [Los autores]

CAPÍTULO 3

DISEÑO Y CONSTRUCCIÓN DEL SISTEMA AUTOPILOTO

3.1. MECÁNICA Y ELECTROMECAÁNICA DE LA AERONAVE

Estructura: La estructura por utilizar consiste en un chasis F450, la cual está fabricada con un material plástico de alta resistencia y que además consta de una power board o placa de conexonado en el centro de esta para poder realizar allí la conexión eléctrica de los variadores. Las piezas que conforman la estructura son mostradas en la Figura 0.1:



Figura 0.1 Piezas del chasis DJI F450 [Los autores]

En primera medida se realiza el ensamble del chasis DJI F450 como se muestra en la Figura 0.2:



Figura 0.2 Ensamble de la estructura DJI F450 [Los autores]

Motores brushless: Este tipo de motores son también llamados como motores sin escobillas y son ampliamente utilizados para diferentes configuraciones de aeronaves multi rotor; estos, a diferencia de los motores eléctricos con escobillas, no presentan tanto desgaste al momento de su funcionamiento.

Para el desarrollo de la aeronave en cuestión se requiere de un total de 4 motores Outrunner 2212 920kv, que se muestran en la Figura 0.3:



Figura 0.3 Motores brushless Outrunner 2212 920kv [Los autores]

Hélices: Para esta primera prueba se utilizan 4 Hélices del DJI genéricas F450, las cuales se muestran en la Figura 0.4:



Figura 0.4 Hélices DJI genéricas F450 [Los autores]

3.2. ELECTRÓNICA DE LA AERONAVE

Controladores de velocidad electrónicos: Los controladores de velocidad (de las siglas en inglés ESC o speed control) como su nombre lo indica, su propósito es variar la velocidad de un motor eléctrico y a su vez establecer el sentido de giro de este según la forma en que sea conectado. Su funcionamiento se rige por la modulación por ancho de pulso o PWM y se ve reflejado cuando el receptor del sistema autopiloto envía dicha señal al módulo ESC con variaciones de tiempo de 1 a 2 milisegundos, donde se tiene que para 1 milisegundo el motor se haya detenido, para 1.5 milisegundos el motor gira a la mitad de su potencia y a 2 milisegundos este funciona al 100% de su potencia.

Dado que existen controladores tanto para motores con o sin escobillas, es de importancia elegir el adecuado ya que no son compatibles entre sí. Para el segundo caso y acorde a los motores previamente elegidos, se tiene que estos se diferencian por traer 3 cables de conexión al motor en vez de 2, motivo por el cual los variadores para este tipo de motores crean una corriente trifásica a partir de la corriente continua proveniente de la batería y que es conducida por los cables en cuestión. Se tiene que uno de los terminales genera un voltaje pequeño que es directamente proporcional a la velocidad de giro del motor más conocida como fuerza electromotriz, este a su vez indica en cualquier instante de tiempo el sentido de giro de dicho motor y con dicha información el controlador es capaz de establecer cómo distribuye la corriente a los imanes del motor para que este gire.

Tras conocidas las características de los motores brushless en el numeral anterior se eligen entonces los controladores de velocidad electrónicos según la corriente que demandan, siendo dichos controladores escogidos con un valor mayor para así evitar posibles contratiempos o accidentes por un pico de corriente que puede llegar a dañar el controlador y hacer que la aeronave caiga en picada. A continuación, en la Figura 0.5 se muestran los controladores utilizados para la aeronave:

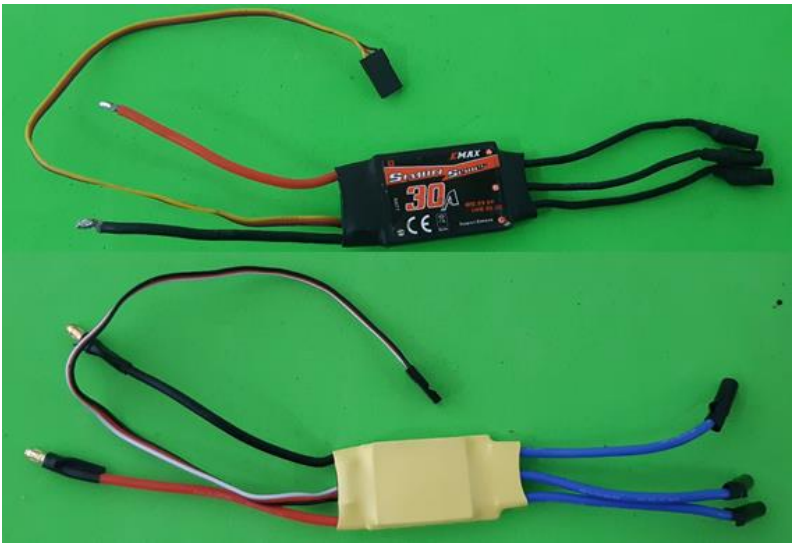


Figura 0.5 ESC's implementados en la aeronave [Los autores]

Aclaración: Los colores del cableado varían según el modelo y marca del ESC, a continuación, en la **¡Error! No se encuentra el origen de la referencia.** se muestran las variantes presentes en el conector de 3 pines según los ESC que son implementados en este proyecto con su respectiva designación:

Variante	Color	Designación
1	Blanco	Señal
	Rojo	Vcc
	Negro	Gnd
2	Amarillo	Señal
	Naranja	Vcc

	Café	Gnd
--	------	-----

Tabla 3.1 Pines de conexión del ESC

Los cables de alimentación del ESC de color rojo y color negro, corresponden a positivo y tierra que van a la salida del módulo de poder de la batería.

Batería: La fuente de alimentación de la aeronave consta de una Batería LiPo (batería de polímero de iones de litio, de ion de litio polímero o más comúnmente batería de polímero de litio) R/C de marca Zippy 2200mah 11.1V 25C. Dicha batería se muestra en la Figura 0.6:



Figura 0.6 Batería Zippy 2200 mAh [Los autores]

Módulo de alimentación 3DR: Se encarga de alimentar al sistema autopiloto y los ESC de forma controlada, ya que suministra información sobre el consumo de corriente y voltaje de la batería de LiPo en tiempo real, de forma que la tarjeta de control pueda calcular el tiempo restante de vuelo acorde a la capacidad restante. Los voltajes de operación máximo y mínimo del módulo corresponden a 18 V y 4.5 V respectivamente. Dicho módulo puede observarse en la Figura 0.7: [30]



Figura 0.7 Módulo de alimentación 3DR [Los autores]

3.3. COMUNICACIONES

GPS: Se utiliza el sistema GPS (de sus siglas en inglés Global Positioning System o sistema de posicionamiento global) que viene preconfigurado para controladores APM y Pixhawk, esta cuenta en el interior de su carcasa con un chip NEO-M8, el cual es de alta sensibilidad y con un tiempo de respuesta de menos de un (1) segundo. El módulo junto con su respectiva estructura se muestra en la Figura 0.8: [31]



Figura 0.8 Módulo GPS y estructura de soporte [Los autores]

Telemetría: El módulo de transmisión de datos de aire y tierra 3DR radio telemetría v3.2 tiene una frecuencia de transmisión de 433 MHz y una potencia de transmisión de 30 DBM (decibelio-milivatio) (1000 mW 9600 bps o bits por segundo), lo que le permite tener un alcance de telemetría de datos de hasta 3 km. Por otro lado, su peso aproximado sin contar la antena es de 11 gr aproximadamente. Este será utilizado para establecer una conexión inalámbrica entre el sistema autopiloto que ejecuta PX4 y la estación de control QGroundControl. El módulo en cuestión que está conformado por un emisor y un receptor se muestra en la Figura 0.9: [32]



Figura 0.9 Módulo de telemetría [Los autores]

PPM Encoder: Este módulo codificador PPM (de sus siglas en inglés Pulse Position Modulation o modulación por posición de pulso) se utiliza para codificar las señales PWM (de sus siglas en inglés Pulse width modulation o modulación por ancho de pulso) provenientes de los canales de un receptor R/C para el control de la aeronave a una sola señal PPM adecuada para el sistema autopiloto

quien cuenta con salidas directas a cada motor. El módulo es mostrado en la Figura 0.10: [33]

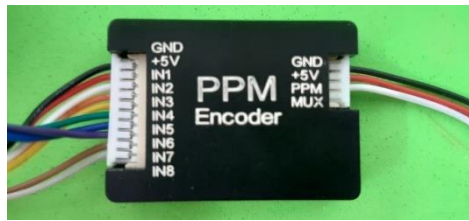


Figura 0.10 PPM Encoder [Los autores]

Control remoto FlySky fs-t6: Es un radio control (transmisor) proporcional computarizado programable de 6 canales a 2.4 GHz que puede ser utilizado para diferentes configuraciones de aeronaves, tales como helicópteros, planeadores y aviones. Este viene junto con su receptor de referencia R6-B. Entre otras características, se encuentra que este control consta de salida PPM, así como un puerto de conexión para ser utilizado en el desarrollo de simulaciones por computadora. El control remoto y su receptor respectivo se muestran en la Figura 0.11: [34]



Figura 0.11 Control remoto FlySky fs-t6 y receptor RB-6 [Los autores]

3.4. ENSAMBLE

En primera estancia, se procede a fijar el módulo GPS en su respectivo soporte mediante un adhesivo doble faz y seguido de esto, tomando como referencia el brazo trasero izquierdo del chasis, se retiran dos de los tornillos y se sitúa allí la base de la estructura del GPS. Para ubicar este módulo, se debe observar en la Figura 0.8 que este trae una flecha marcada en su carcasa, la cual es de vital importancia ya que ésta debe coincidir con la flecha blanca del sistema autopiloto. El acople del GPS queda según lo mostrado en la Figura 0.12:



Figura 0.12 Fijación del módulo GPS [Los autores]

A continuación, se ensambla la estructura antivibración para fijar allí el sistema autopiloto haciendo uso de adhesivos doble faz. Posteriormente se ubica y se fija dicha estructura en la parte central superior del chasis con amarras plásticas, de forma que la flecha en la carcasa del Pixhawk coincida con el frente de la aeronave. El proceso llevado a cabo se muestra en la Figura 0.13:



Figura 0.13 Acople del sistema autopiloto [Los autores]

Ahora se procede a ubicar los motores brushless acorde al sentido de giro de estos tomando como referencia la Figura 0.14. Otra referencia importante es el color de la tuerca que viene con cada motor, ya sea negra o plateada, ya que esto indica que el motor está diseñado de fábrica para girar en sentido antihorario (ccw) u horario (cw) respectivamente. Los motores tras ser ubicados se muestran en la Figura 0.15:

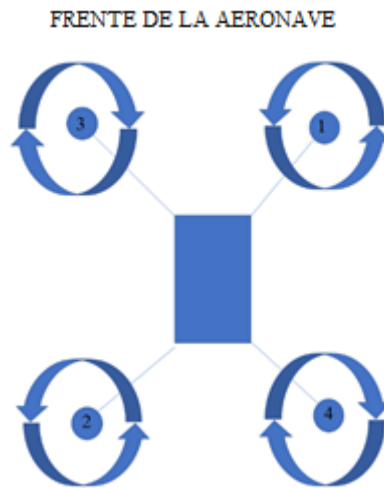


Figura 0.14 Sentido de giro de los motores [Los autores]



Figura 0.15 Acople de los motores brushless [Los autores]

Seguido de ello se ubica cada uno de los ESC en cada brazo de la estructura de la aeronave como se aprecia en la Figura 0.16:



Figura 0.16 Acople de los ESC's [Los autores]

La siguiente etapa consiste en ubicar el buzzer (zumbador), el cual se sitúa lo más lejos posible del sistema autopiloto para evitar así posibles interferencias; adicional a este se acopla también el interruptor de seguridad, usando la estructura del GPS como soporte. En la Figura 0.17 se muestra la ubicación de los elementos en cuestión en la estructura de la aeronave:



Figura 0.17 Acople del buzzer y del interruptor de seguridad [Los autores]

A continuación, se ubica el módulo de telemetría en la parte derecha justo a un lado del sistema autopiloto, como se observa en la Figura 0.18:



Figura 0.18 Acople del módulo de telemetría [Los autores]

Tras lo anterior, se toma el módulo PPM Encoder y el receptor del control remoto, fijando estos con una amarra plástica en la parte izquierda, al lado del sistema autopiloto. Adicionalmente, se usa el soporte del GPS para fijar allí la antena del receptor, como se muestra en la **¡Error! No se encuentra el origen de la referencia.:**

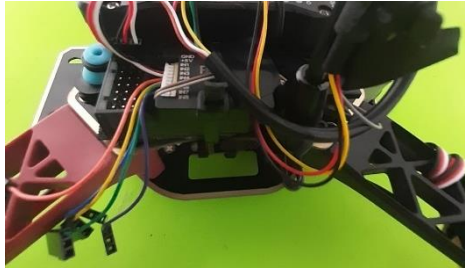


Figura 0.19 Fijación del PPM Encoder y el receptor rc [Los autores]

Finalmente, se ubica la batería en el espacio intermedio existente en la estructura de la aeronave, garantizando que esta quede firme pero que a su vez pueda ser extraída con facilidad para cuando deba realizarse la etapa de carga de esta. Esta es ubicada como se observa en la Figura 0.20:



Figura 0.20 Ubicación de la batería [Los autores]

3.5. AUTOPILOTO DE LA AERONAVE

3.5.1. Conexión de sensores y periféricos

En primera estancia se conectan el buzzer (zumbador) y el interruptor de seguridad. Los puertos para su conexión y su respectivo proceso se muestran en la Figura 0.21:

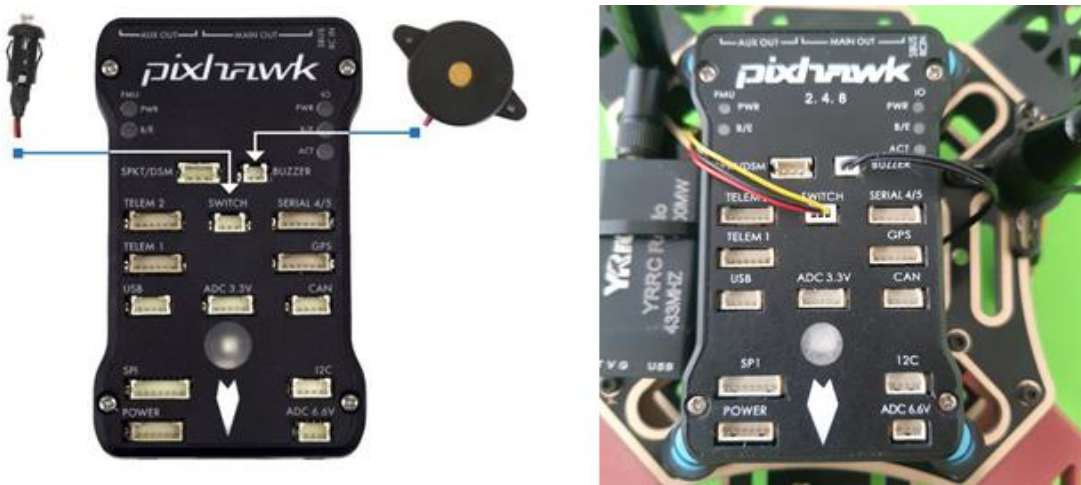


Figura 0.21 Puertos de conexión del interruptor de seguridad y el buzzer [Los autores]

Pasando ahora al módulo GPS, se encuentra que este cuenta con una brújula la cual se conecta vía i2c, sin embargo, vale aclarar que el Pixhawk cuenta también con una brújula interna que puede utilizarse de ser necesario.

La conexión de este módulo también es bastante simple. Se toma el conector de 6 pines y se inserta en el demarcado como GPS, el conector restante que es de 4 pines por otro lado se inserta en el demarcado como i2c. El proceso se lleva a cabo según la Figura 0.22 :



Figura 0.22 Conexión del módulo GPS [Los autores]

Ahora, se procede a conectar el módulo de telemetría, haciendo uso del puerto designado como TELEM1, como se muestra en la Figura 0.23:



Figura 0.23 Conexión del módulo de telemetría [Los autores]

3.5.2. Conexión y prueba de los motores

A continuación, ciñéndose a lo descrito en la Figura 0.14, se procede a conectar y ensayar de manera independiente cada motor brushless. Para ello, se

conectan las terminales de los motores según el caso a lo mostrado en la Figura 0.24:

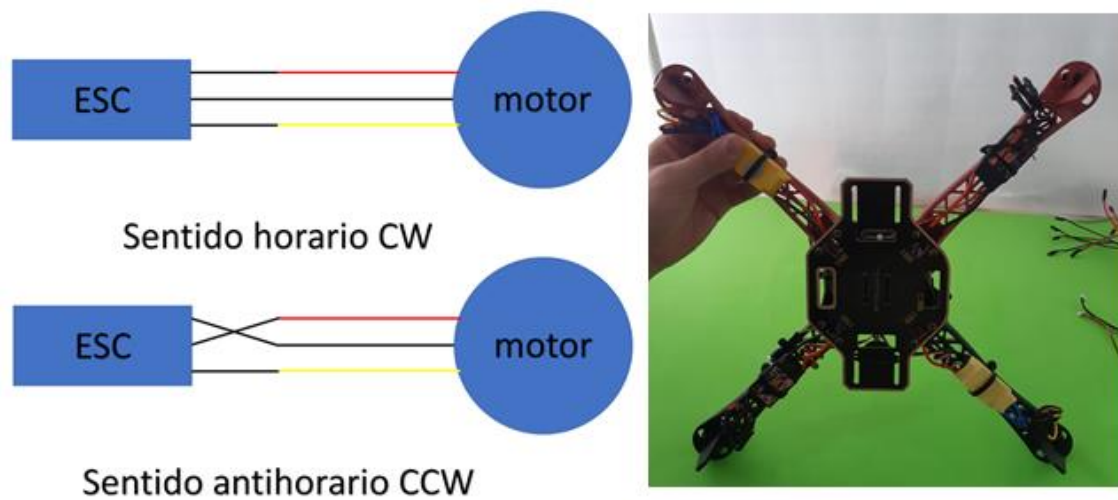


Figura 0.24 Conexión de los ESC [Los autores]

Luego, se procede a conectar el terminal de tres pines saliente del ESC al canal 3 (CH3) del módulo receptor del control remoto, de forma que el cable de color más oscuro (para este caso café o negro) apunte hacia la parte exterior de dicho módulo. Esto es de vital importancia ya que de allí viene la alimentación del receptor y si se invierte la polaridad el módulo se daña inmediatamente.

Tras ello, se enciende el emisor (control remoto) y finalmente se conectan los terminales de alimentación del ESC a la batería a través del módulo de alimentación respectivo quien actúa a su vez como elemento de protección. Con lo realizado en el paso anterior, el ESC emitirá un sonido y el motor llevará a cabo movimientos muy leves mientras es configurado.

Ahora se realiza la prueba de encendido del motor brushless, donde para ello se acciona la palanca izquierda del control remoto simulando el despegue de la aeronave, con ello el motor debe empezar a girar y se ha de verificar que su sentido de giro sea el adecuado. Nota: esta prueba debe realizarse sin las hélices puestas. A continuación, en la Figura 0.25 se muestra la etapa de conexión, alimentación y prueba de uno de los motores:

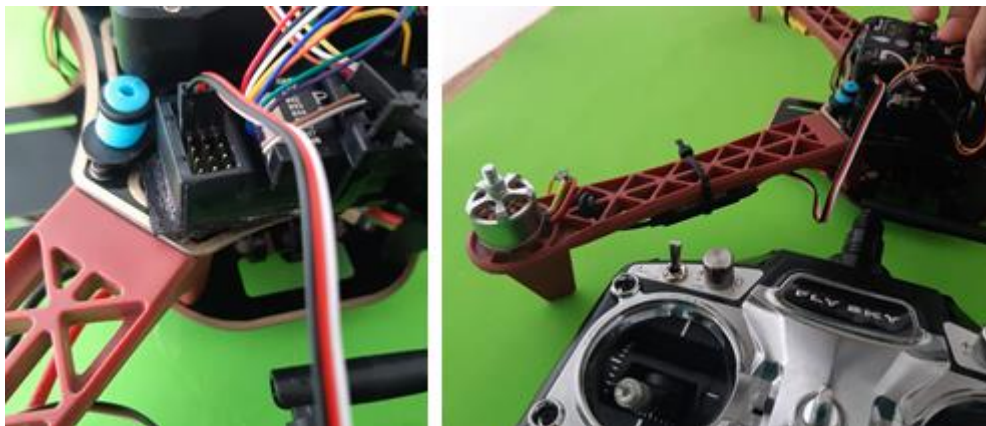


Figura 0.25 Conexión y prueba de los motores [Los autores]

Tras haber realizado la prueba de los motores, se pasa entonces a conectar los ESC directamente al sistema autopiloto; para ello basta con tener en cuenta la numeración establecida para cada motor previamente. Se debe tener en cuenta que el cable correspondiente a tierra (oscuro) del conector de tres pines debe quedar hacia arriba. El proceso se lleva a cabo como se observa en la Figura 0.26:



Figura 0.26 Conexión de pines de control de los ESC's [Los autores]

El sistema autopiloto Pixhawk 2.4.8 cuentan con una entrada preestablecida para el receptor, sin embargo, el correspondiente radio control FlySky FS-T6 no puede ser conectado directamente, por lo que se requiere implementar el módulo PPM Encoder. Para ello, se toma e inserta cada conector según número del canal y pin respectivo. Como se puede apreciar en la Figura 0.27, los canales 7 y 8 no se conectan, esto dadas las características propias del radio control.

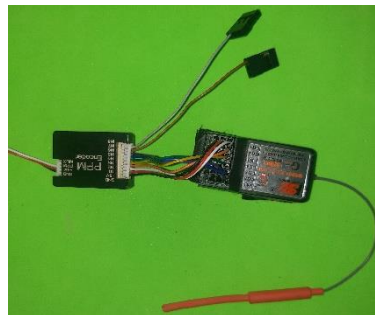


Figura 0.27 Conexión del receptro R/C al PPM Encoder [Los autores]

Finalmente, según la Figura 0.28, se conecta el cableado saliente del módulo a la entrada denominada como "RC", teniendo en cuenta que el cable de color negro debe quedar en la parte superior.

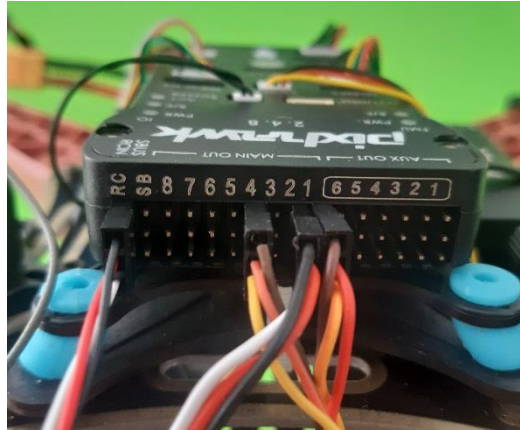


Figura 0.28 Conexión del PPM Encoder al sistema autopiloto [Los autores]

3.6. PROGRAMACIÓN, SIMULACIÓN Y CALIBRACIÓN DEL AUTOPILOTO

3.6.1. Carga del firmware

Para la realización de la carga del firmware PX4 se abre el programa QGroundControl sin tener conectado aún el sistema autopiloto a la computadora, seguido de ello se va a la pestaña “Vehicle Setup” y posteriormente a “Firmware”. El programa indica que se debe conectar vía cable USB, y tras ello se elige en la parte derecha entre PX4 Pro o ArduPilot, siendo la primera la opción a seleccionar, como se muestra en la Figura 0.29:

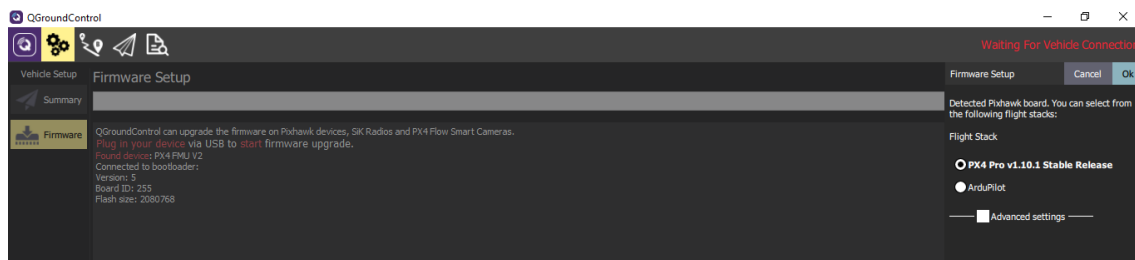


Figura 0.29 Selección y carga del firmware [Los autores]

Finalmente se presiona “Ok” y se espera a que culmine el proceso (esto será indicado tanto en pantalla como en una secuencia de sonido emitida por buzzer del Pixhawk). El resultado a obtener al acabar el proceso se muestra en la Figura 0.30:

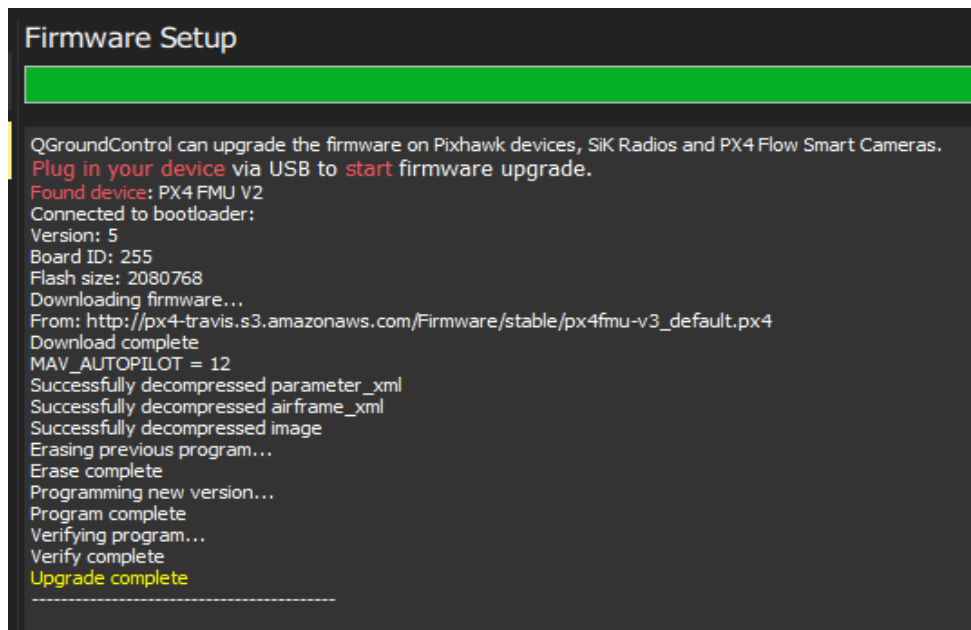


Figura 0.30 Proceso de carga del firmware y verificación [Los autores]

Tras haber realizado el proceso de carga del firmware, debajo de dicha opción se habilitarán los diferentes parámetros a configurar.

3.6.2. Elección de la estructura de la aeronave

Dado que en la primera etapa de este proyecto se busca obtener un despegue vertical de la aeronave, se parte de una estructura estándar de Quadrotor DJI F450 w / DJI ESCs, que se ubica en la lista desplegable de Quadrotor X como se muestra en la Figura 0.31:

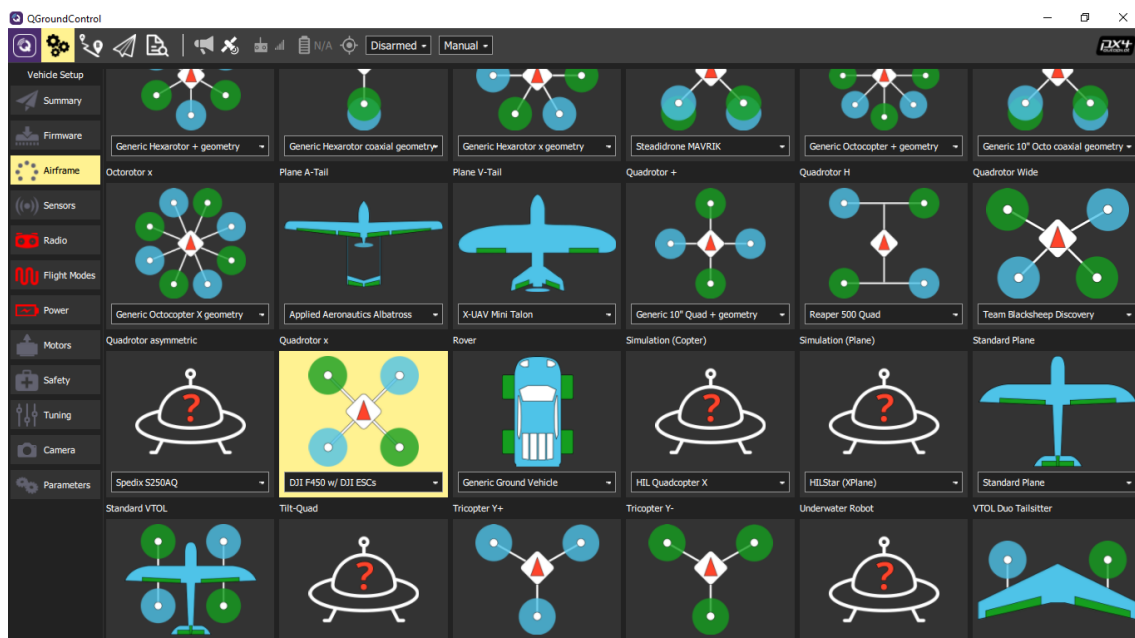


Figura 0.31 Elección de la estructura de la aeronave [Los autores]

Finalmente se va a la parte superior derecha se elige “Apply an restart”, se da “Apply” y se espera a que se cargue la configuración al Pixhawk. Como se observa en la Figura 0.32, tras presionar la opción “summary”, se muestran

diferentes recuadros donde se indican los parámetros configurados y los restantes.



Figura 0.32 Sumario de parámetros configurados y restantes [Los autores]

3.6.3. Configuración de los sensores

Para proceder con la configuración de los diferentes sensores del sistema autopiloto, se selecciona la opción “Sensors”, que se encuentra en el menú lateral que se despliega al elegir previamente “Vehicle Setup”.

Brújula: Para la brújula o compass, el proceso de calibración configura todos los magnetómetros internos y externos conectados. Se elige entonces la opción “Compass”, tras ello se presiona “Ok”. A continuación, se siguen los siguientes lineamientos:

“Coloque el vehículo en cualquiera de las orientaciones que se muestran en rojo (incompleto) y manténgalo quieto. Una vez que se le solicite (la imagen de orientación se vuelve amarilla), gire el vehículo alrededor del eje especificado en ambas direcciones. Una vez que se complete la calibración para la orientación actual, la imagen asociada en la pantalla se volverá verde.” En la Figura 0.33 se muestran las diferentes orientaciones mencionadas, seguido de la Figura 0.34 que indica que el procedimiento fue completado exitosamente.

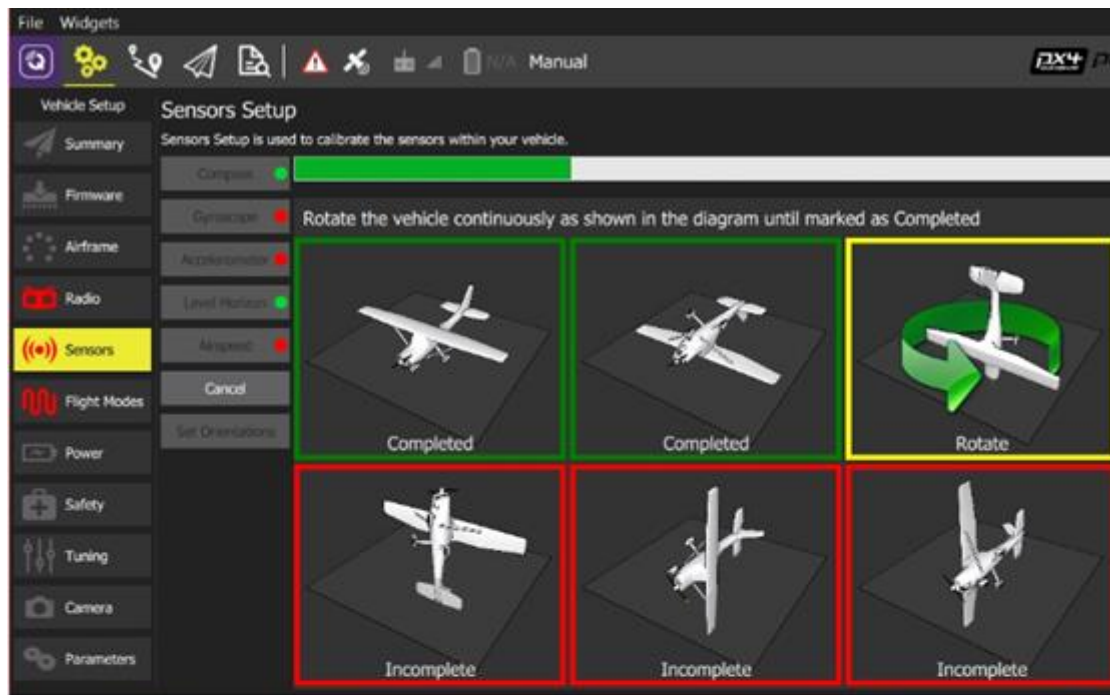


Figura 0.33 Calibración de la brújula (Compass) [Los autores]

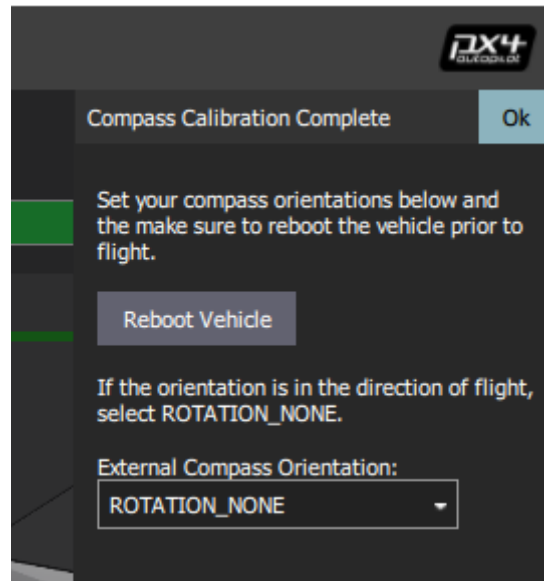


Figura 0.34 Confirmación de calibración exitosa de la brújula (Compass) [Los autores]

“Una vez que haya calibrado el vehículo en todas las posiciones, QGroundControl mostrará Calibración completa (todas las imágenes de orientación se mostrarán en verde y la barra de progreso se llenará por completo). Luego puede pasar al siguiente sensor.”

Giroscopio (Gyroscope): En la lista de sensores se elige “Gyroscope”, y se sitúa la aeronave en una superficie plana de forma que se quede estable y quieto. Seguido de ello presione Ok para comenzar el proceso de calibración. El programa le indicará cuando haya finalizado el proceso, tal como lo indica la Figura 0.35.

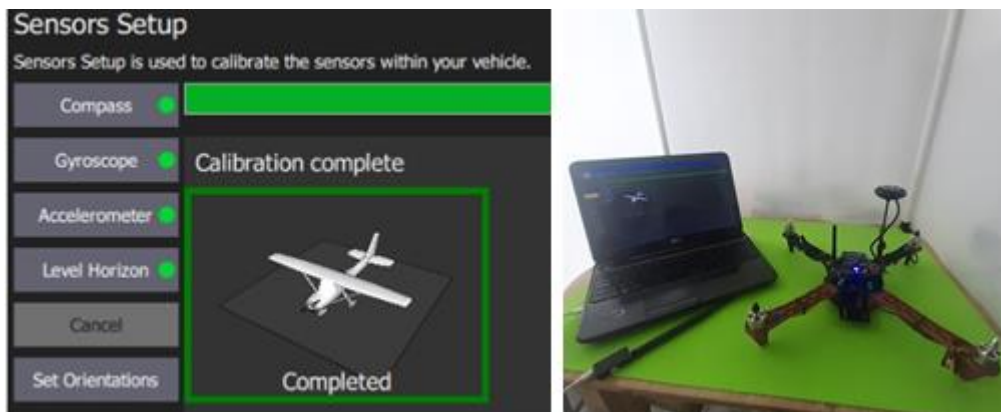


Figura 0.35 Confirmación de calibración exitosa del giroscopio [Los autores]

Acelerómetro (Accelerometer): En la lista de sensores se elige “Accelerometer”. Seguido de ello presione Ok para comenzar el proceso de calibración. A continuación “coloque el vehículo como lo guían las imágenes en la pantalla. Una vez que se le solicite (la imagen de orientación se vuelve amarilla), mantenga el vehículo quieto. Una vez que se complete la calibración para la orientación actual, la imagen asociada en la pantalla se volverá verde.”. La Figura 0.36 muestra las diferentes posiciones en las que se debe situar la aeronave:

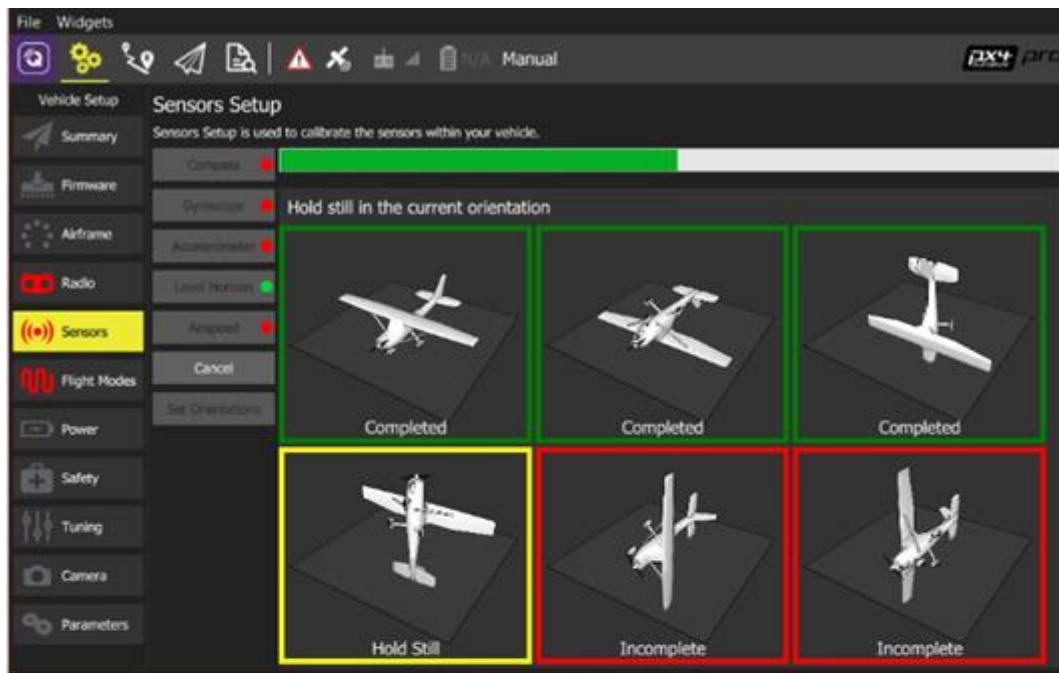


Figura 0.36 Calibración del acelerómetro [Los autores]

“Nivelar Horizonte” (Level Horizon): Este proceso se realiza con el fin de “compensar pequeñas alineaciones erróneas en la orientación del controlador y nivelar el horizonte en la vista de vuelo”.

En la lista de sensores se elige “Level Horizon”. A continuación, coloque la aeronave en su orientación de vuelo nivelado en una superficie nivelada y presione Ok, como se muestra en la Figura 0.37:

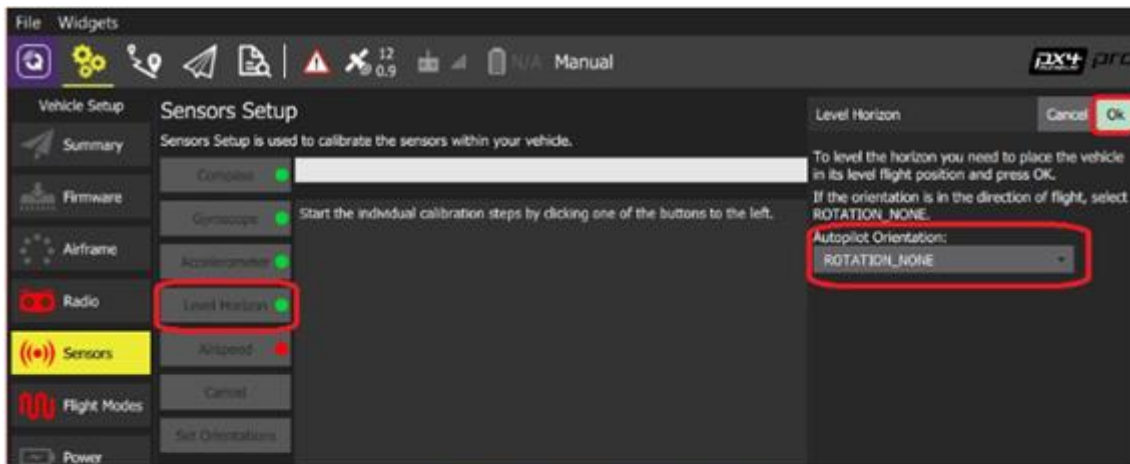


Figura 0.37 Nivelación del horizonte del sistema autopiloto [Los autores]

3.6.4. Calibración del radio control

A continuación, se enciende el radio control y se pasa a la pestaña de “Radio”. En la parte derecha, se selecciona el modo 2, y tras ello se le da a la opción “Calibrate”, donde se desplegará un mensaje en la parte superior derecha, indicando que antes de comenzar el proceso todas las palancas de control, switch y perillas deben estar en posición de cero. Tras lo anterior se presiona “OK”, como se muestra en la Figura 0.38:

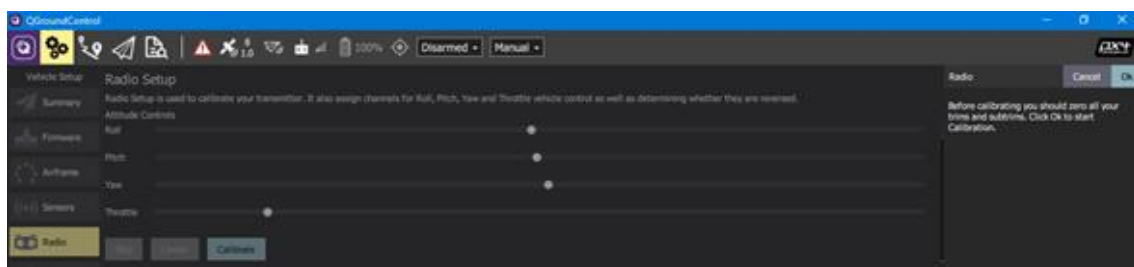


Figura 0.38 Preparación del radio control para su calibración [Los autores]

Ahora se pone la palanca de “Throttle” del radio control en el límite inferior y se oprime “NEXT”, como se indica en la Figura 0.39:

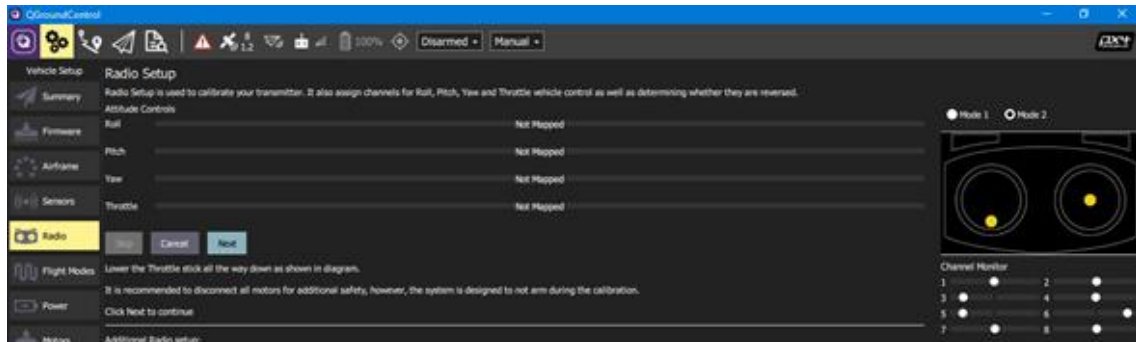


Figura 0.39 Inicio del proceso de calibración [Los autores]

Tras ello, el programa indica que se deben mover las palancas según lo descrito por la gráfica que se muestra en la parte derecha de la pantalla; teniendo así una secuencia que corresponde a las posiciones que se observan en la Figura 0.40 y que estas corresponden a movimientos de Throttle, yaw, pitch y roll, que están ligados a los canales 1, 2, 3 y 4.

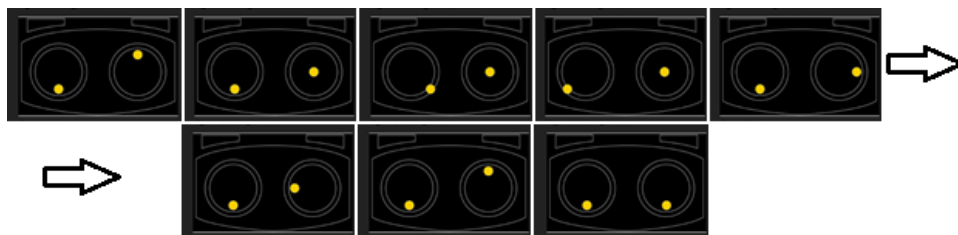


Figura 0.40 Secuencia de posiciones para calibración de movimiento [Los autores]

Tras lo anterior, se debe ahora mover las perillas y demás switches del control remoto a sus posiciones máximas y mínimas en repetidas ocasiones, finalmente se le da a “NEXT” como se muestra en la Figura 0.41:

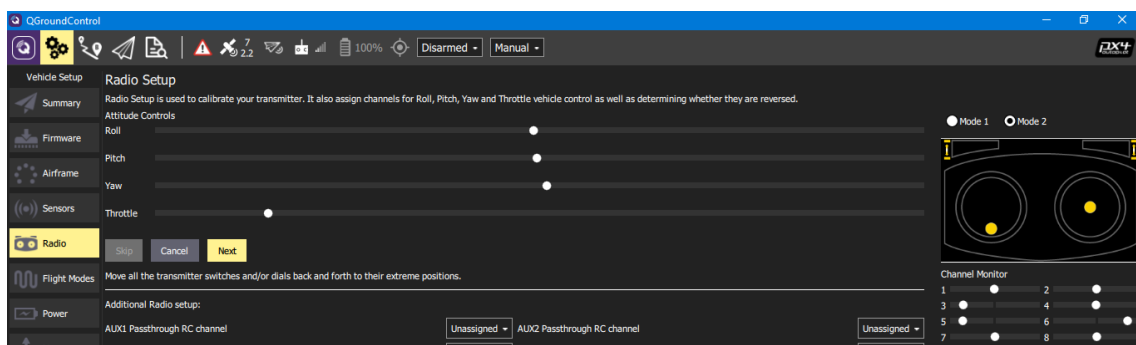


Figura 0.41 Calibración de perillas y switches [Los autores]

A continuación se mostrará un mensaje indicando que se debe presionar de nuevo la opción “NEXT” para guardar los nuevos parámetros de configuración en el sistema autopiloto, tal como se muestra en la Figura 0.42:

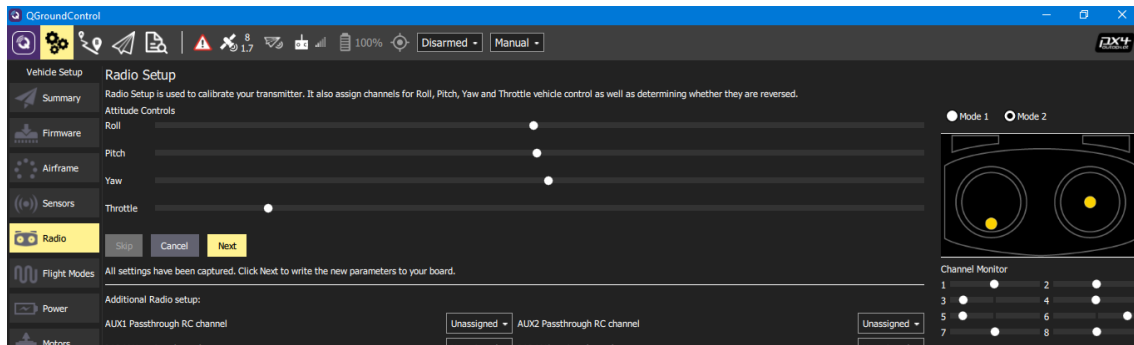


Figura 0.42 Escritura de los nuevos parámetros en la memoria del sistema autopiloto [Los autores]

3.6.5. Configuración de los modos de vuelo

Para configurar los modos de vuelo de la aeronave se requiere configurar uno de los 6 canales para poder así intercambiarlos mediante el accionamiento ya sea de una perilla o uno de los switches y dado que los canales 1, 2, 3 y 4 son exclusivos para el manejo de la aeronave, se debe asignar ya sea el canal 5 o el canal 6 a uno de dichos elementos, lo cual se hace directamente en el menú de funciones del control remoto tal como se indica a continuación:

- Se presiona la rueda situada en la parte inferior izquierda del control.
- Se elige la opción de “Functions setup” y se presiona la tecla “ok”.
- De la lista, se escoge ahora la opción de “Aux. channels” se presiona la tecla “ok”.
- Para el canal 5, se mueve la rueda hasta que indique “SWA”, que viene siendo el switch superior derecho ubicado al lado de la perilla y finalmente se oprime la tecla “ok”.
- Finalmente, se presiona la tecla de “cancel” hasta volver a la pantalla principal.

En la Figura 0.43 se muestra el desarrollo del procedimiento mencionado anteriormente:



Figura 0.43 Configuración del canal 5 en el control remoto [Los autores]

Pasando ahora a la pestaña de “Flight modes”, en la opción “Mode Channel” se elige “Channel 5”, el cual fue configurado en el control remoto anteriormente. Adicionalmente, en la opción “Flight Mode 1”, dejando los demás modos de vuelo sin configurar. Los parámetros quedar ajustados según lo que se muestra en la Figura 0.44:

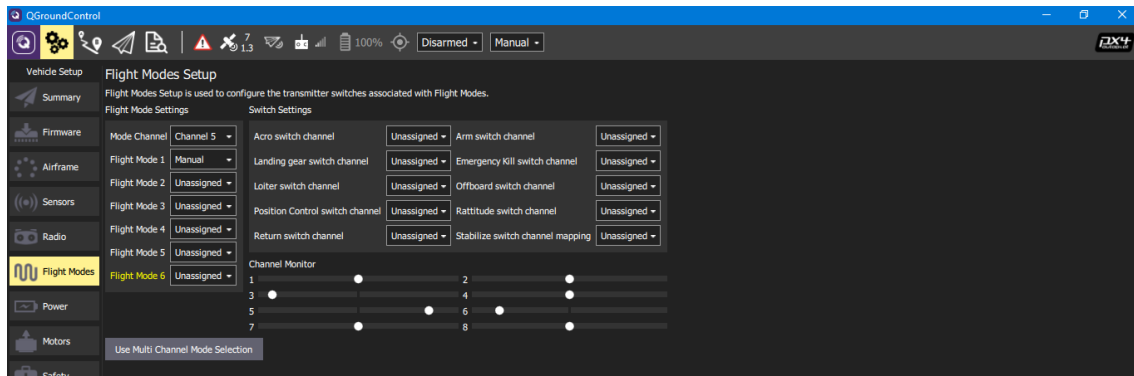


Figura 0.44 Configuración de los modos de vuelo [Los autores]

3.6.6. Ajuste de parámetros de la batería

Según las características propias de la batería implementada para alimentar los diferentes componentes de la aeronave, se deben ajustar ciertos parámetros en la pestaña “power”, de manera que queden como lo mostrado en la Figura 0.45:

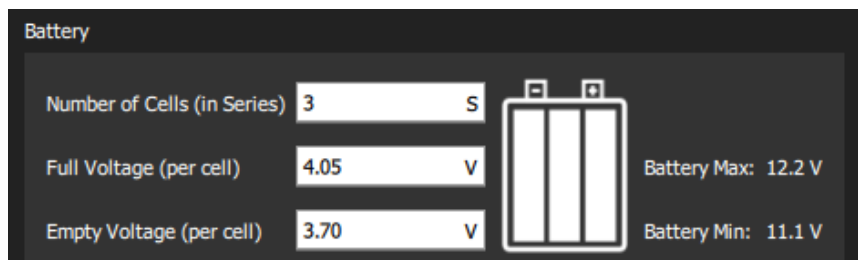


Figura 0.45 Ajuste de parámetros de la batería [Los autores]

3.6.7. Configuración del parámetro COM_POWER_COUNT

Este parámetro se utiliza para verificar que el número esperado de canales de alimentación de 5V esté debidamente conectado. Dado que el sistema autopiloto Pixhawk 2.4.8 solo cuenta con un puerto de alimentación a parte de la conexión de los ESC, se debe poner el parámetro en cuestión en cero (0), que equivale en realidad a solo un solo canal dada la versión de firmware implementada. Para poder realizar dicho cambio, se va a la pestaña “Parameters” ubicada en la parte inferior izquierda dentro del menú “Vehicle Setup”, como se observa en la Figura 0.46 en la parte superior aparece un buscador, donde se escribe el comando en cuestión y se cambia el valor de fábrica que es de 1 a 0.

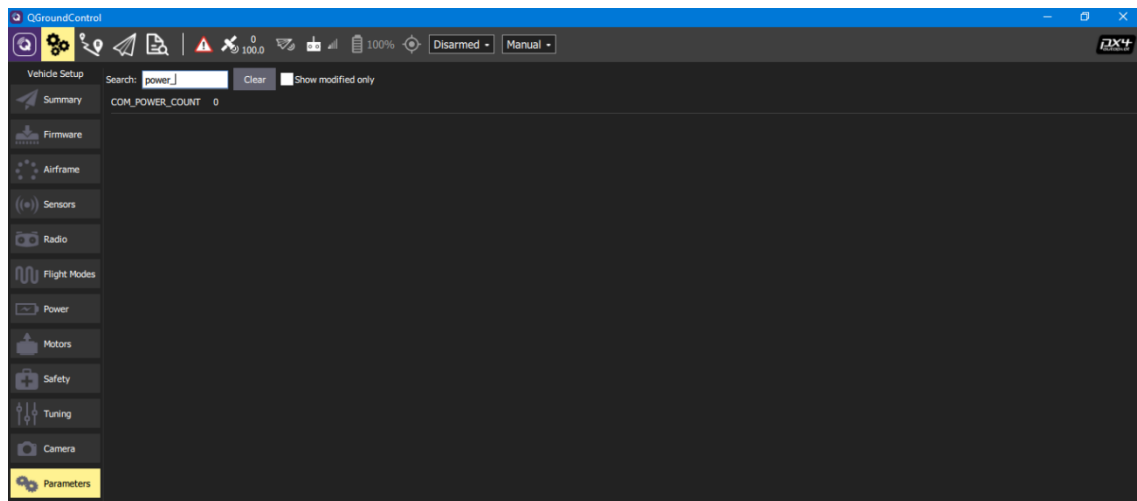


Figura 0.46 Configuración del parámetro COM_POWER_COUNT [Los autores]

CAPÍTULO 4

ENTRENAMIENTO MEDIANTE SOFTWARE DE AEROMODELISMO

Antes de proceder con la realización de pruebas de vuelo en un entorno real, es de utilidad hacer uso de herramientas de simulación que permitan a la persona comprender como se comporta y se controla la aeronave según el accionamiento de los diferentes mandos del radio control. Para ello, se utiliza el simulador de vuelo Phoenix R/C, donde se utilizará un modelo de QuadCopter incluido por defecto en las librerías de dicho programa. Por otro lado, para poder llevar a cabo simulaciones en este software, se requiere de un radio control con su respectivo conector MIDI a USB el cual se encarga de enviar las señales PPM a través de dicho medio; no obstante, ya que no se cuenta con este, se opta por usar un dispositivo Android, el cual, mediante un servidor y un joystick virtual, se comunicará con el simulador haciendo de transmisor y permitiendo así controlar la aeronave.

4.1. CONFIGURACIÓN DEL JOYSTICK

4.1.1. Instalación del Launcher para la detección del mando

En primera medida, tras haber instalado el simulador, se descarga el archivo “PhoenixRC_emu_v0_3”, el cual se debe descomprimir y posteriormente copiar los archivos obtenidos en el directorio C:\Program Files (x86)\PhoenixRC, adicionalmente se deben brindar los permisos solicitados y renombrar el archivo “PhoenixRC.exe” a “update.exe”, seguido de ello, se debe renombrar el archivo “simulator.exe” a “PhoenixRC.exe”. Tras lo anterior, se da click derecho a “launcher.exe” luego a “propiedades” y luego a “compatibilidad”, donde se debe marcar la casilla de “ejecutar este programa como administrador” y así finalmente a “aplicar” y luego a aceptar. El procedimiento y los resultados finales a obtener se muestran en la Figura 0.1:

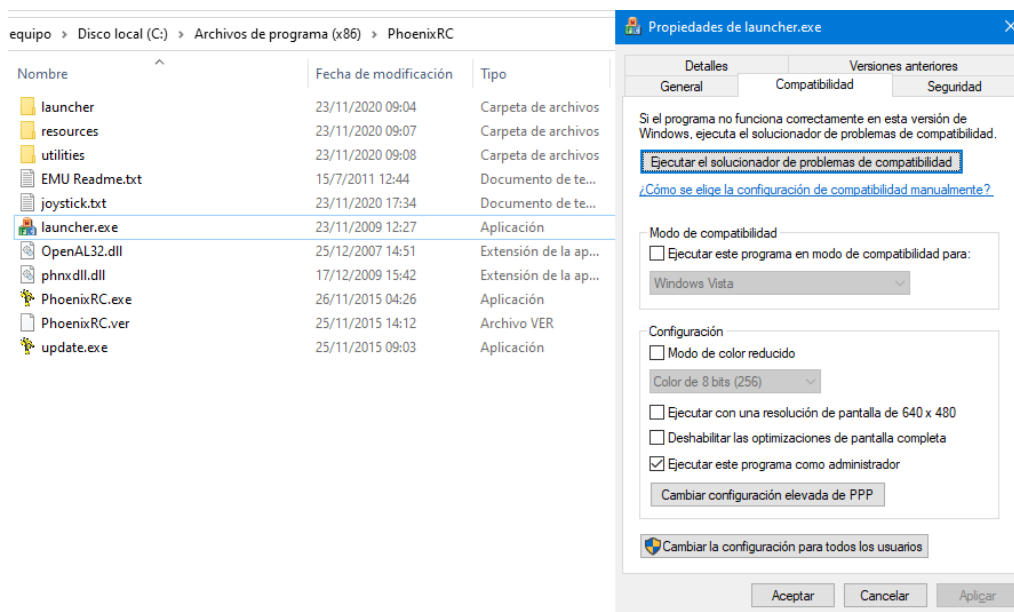


Figura 0.1 Instalación y configuración del launcher [Los autores]

4.1.2. Configuración de la aplicación y conexión al servidor

Se debe ejecutar primero la aplicación DroidJoy server, y tras ello abrir la aplicación móvil, donde se debe elegir la opción “Connect” y posteriormente en “Search server”, donde debe aparecer la dirección IP mostrada en el servidor, la cual debe ser seleccionada y nombrada para así poder luego elegir en la lista el servidor y entablar una conexión vía Wifi con la computadora. El proceso se muestra en la Figura 0.2:

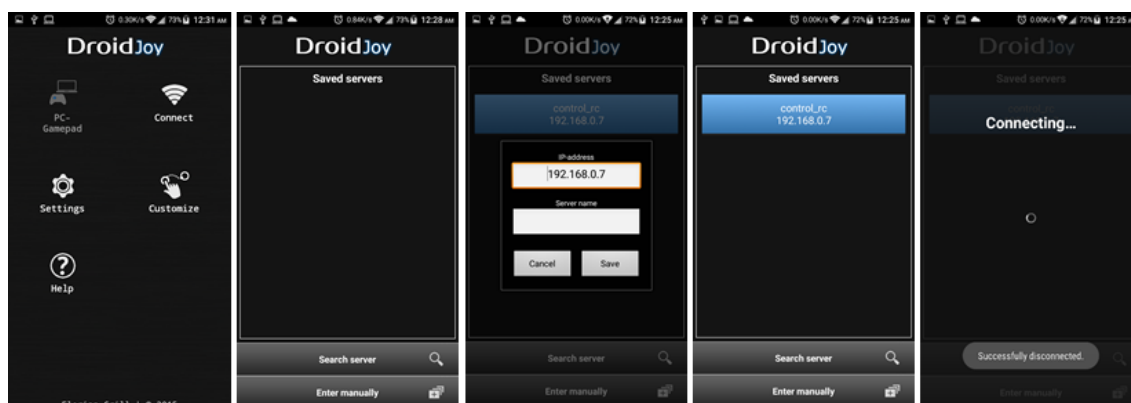


Figura 0.2 Conexión entre el dispositivo Android y el servidor [Los autores]

A continuación, en el menú principal se elige la opción de “Customize”, seguida de la opción “New profile” para crear un mando personalizado como se muestra en la Figura 0.3:

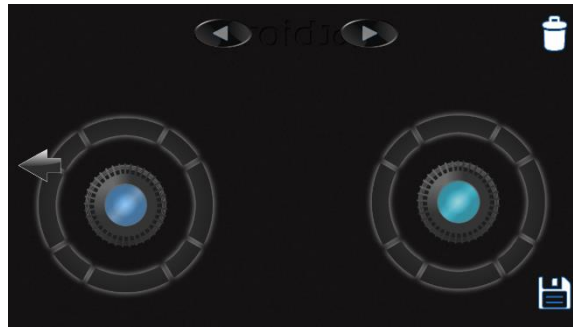


Figura 0.3 Perfil de mando personalizado [Los autores]

Finalmente, en el menú principal se escoge “PC-Gamepad” y de la lista que se muestra se escoge el perfil creado previamente.

4.1.3. Configuración del mando en el software de simulación

Pasando ahora al computador, se procede a ejecutar “launcher.exe”, donde aparecerá una lista desplegable en la que debe aparecer “Microsoft PC-joystick driver” que representa el mando creado anteriormente. A continuación, se presiona “Run Phoenix” para ejecutar el simulador, donde en primera estancia se aparecerá el menú de inicio mostrado en la Figura 0.4:

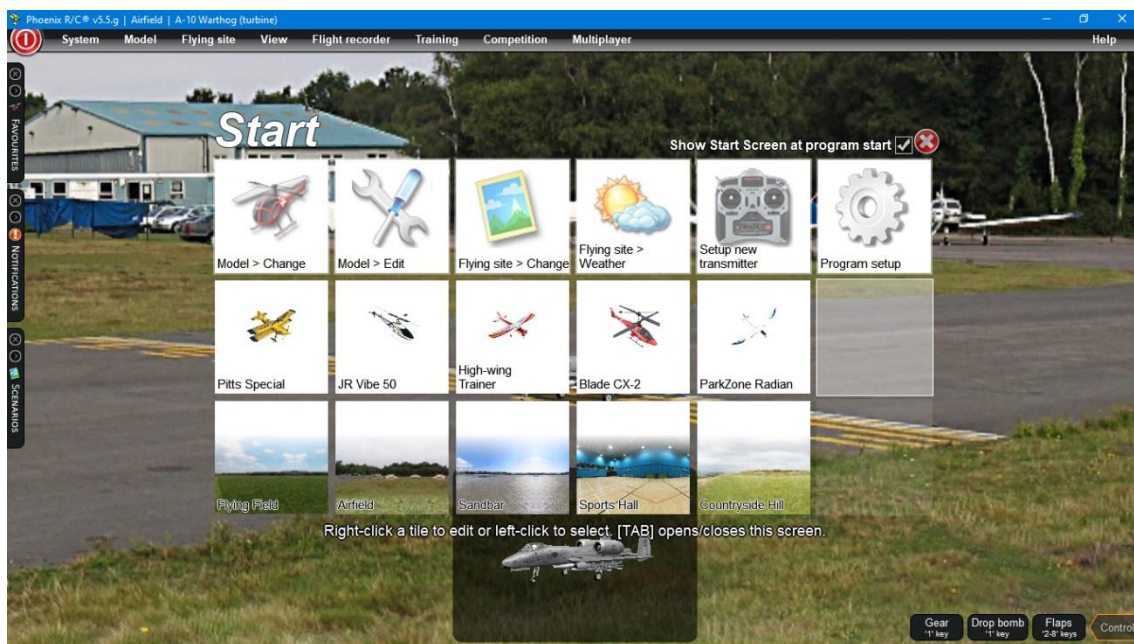


Figura 0.4 Menú de inicio de Phoenix R/C [Los autores]

Ahora, se escoge la opción de “Setup new trasnmmitter”, que nos llevará a un asistente para configurar el radio control, que en este caso será emulado por el dispositivo Android. Se debe seguir cada paso descrito, ya que de ello depende una correcta configuración del radio control; en primera medida se llevan a cabo las preparaciones preliminares de la Figura 0.5:



Figura 0.5 Preparaciones preliminares [Los autores]

Tras lo anterior, se colocan las palancas de mando en su posición central, es decir, se dejan sin accionar, y seguido de ello, cuando aparecen unas barras demarcadas con los diferentes canales, se procede a mover dichas palancas en diferentes direcciones de forma que sean reconocidas por el programa como se muestra en la Figura 0.6:



Figura 0.6 Detección y calibración de las palancas del mando [Los autores]

A continuación, se deben presionar los botones creados en la parte superior del mando, ya que estos actuarán como si fueran los switch de este. Después de ello se presiona “Finish”, como se observa en la Figura 0.7:



Figura 0.7 Detección y calibración de los switchs del mando [Los autores]

El siguiente paso consiste en crear un nuevo perfil para el mando en cuestión. Para ello, en el asistente en pantalla se elige la opción “Custom” y tras presionar “Next” dos veces, se le debe asignar ahora un nombre y en la parte inferior se debe marcar la opción “Quick setup”, luego se deben dejar las palancas del mando en su posición por defecto (en el centro). Las indicaciones se muestran en la Figura 0.8:



Figura 0.8 Creación del perfil para el mando [Los autores]

Para evitar posibles errores al configurar el mando, se le da a “Next” hasta que el asistente indique que el proceso ha finalizado. Ahora, se procede a asignar los canales de forma manual, y para ello se va al menú superior izquierdo, se selecciona “System” seguido de “Your Controls”, donde se debe seleccionar el perfil creado y posteriormente presionar “Edit”. Esta etapa se muestra en la Figura 0.9:

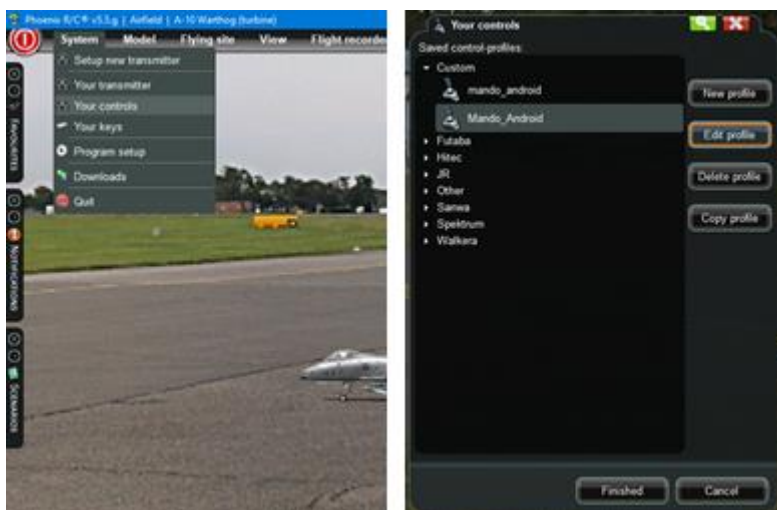


Figura 0.9 Selección del mando a configurar [Los autores]

Ahora se deben asignar los canales correspondientes en invertirlos de ser necesario, lo cual debe quedar configurado acorde a la Figura 0.10. Posteriormente se presiona “Finished”:

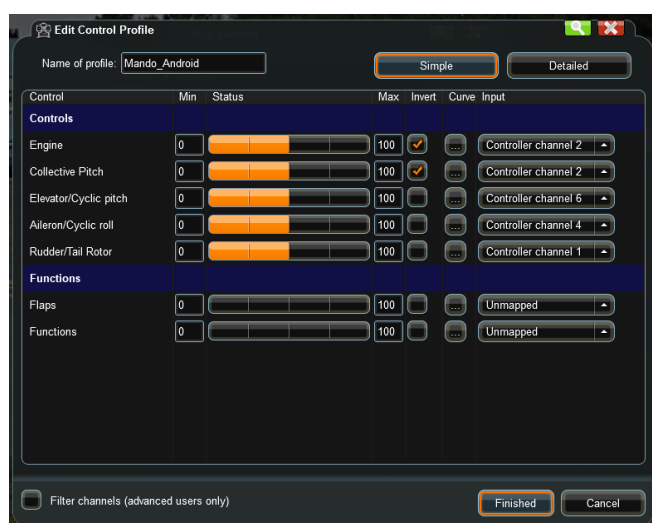


Figura 0.10 Asignación e inversión de canales [Los autores]

4.2. Elección de la aeronave

Para esta parte, se va a la pestaña “Model” que está en la parte superior izquierda de la pantalla y seguido de esto se elige la opción de “Change”. Con lo anterior, se abrirá un menú que cuenta con una lista en la parte izquierda y una imagen del modelo actual seleccionado en la parte derecha. Se debe buscar entonces el desplegable de “Multi-rotors” y escoger de allí el modelo “DJI Phantom” dándole doble click. Para culminar el proceso se le da en “Finished”, con lo cual el modelo seleccionado aparecerá ahora en el entorno de simulación. Este proceso se muestra en la Figura 0.11:



Figura 0.11 Elección de la aeronave DJI Phantom [Los autores]

4.3. Activación de los binoculares

Para poder visualizar de forma más clara la actitud de la aeronave al momento del vuelo, se puede habilitar la opción de “Binoculars”, la cual permite visualizar un recuadro en la parte central inferior de la pantalla donde se muestra la aeronave en primera persona. Para realizar esto, se va al menú superior izquierdo y se elige la pestaña “View” seguido de la opción “Displays”, donde se despliega una lista de opciones en la que se marca “Binoculars”, finalizando así la configuración. El proceso llevado a cabo se muestra en la Figura 0.12:

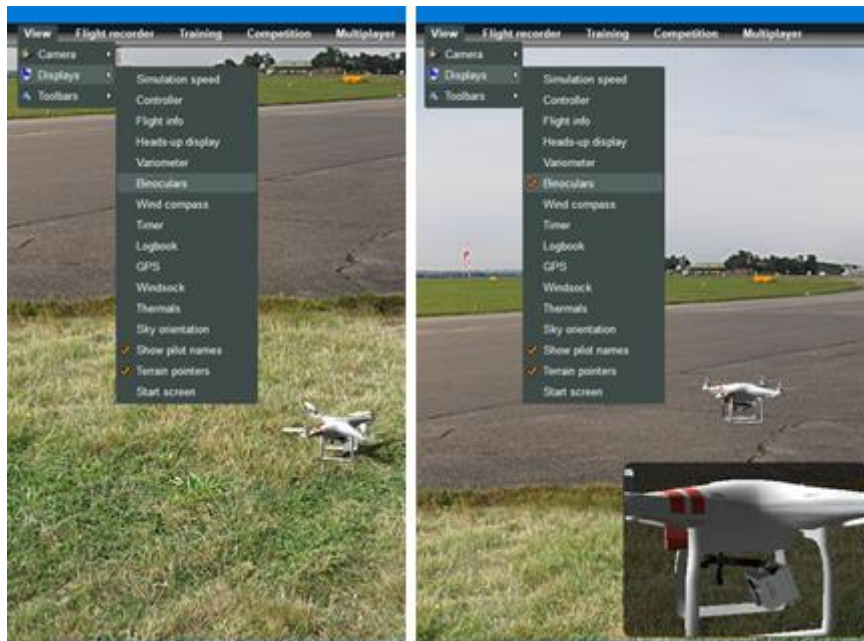


Figura 0.12 Activación de los binoculares [Los autores]

4.4. Desarrollo de simulaciones

Dadas las características del simulador, se tiene entonces un entorno en el cual se puede ensayar y aprender las nociones básicas del control de aeronaves por medio de radio control, teniendo la posibilidad de cometer errores sin entrar en daños materiales físicos. Se encuentra entre otras opciones, que en caso de que la aeronave colisione ya sea en una superficie sólida o en agua, el mismo simulador se encarga de mostrar dichos daños y de reiniciar la simulación en el punto de partida inicial. A continuación, en las Figura 0.13 y Figura 0.14 se muestran capturas de simulaciones llevadas a cabo en un campo abierto y al lado de un río en medio de la ciudad. Para cambiar el escenario de vuelo se va a la pestaña “Flying site” y posteriormente a “Change”, donde aparecerán diferentes entornos.

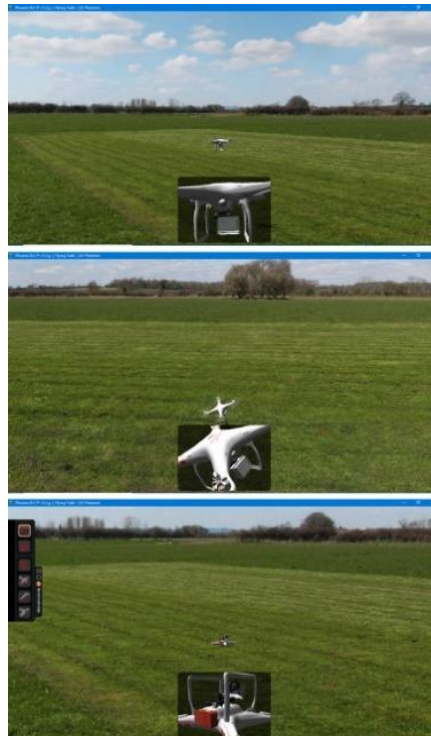


Figura 0.13 Simulación de vuelo en campo abierto [Los autores]

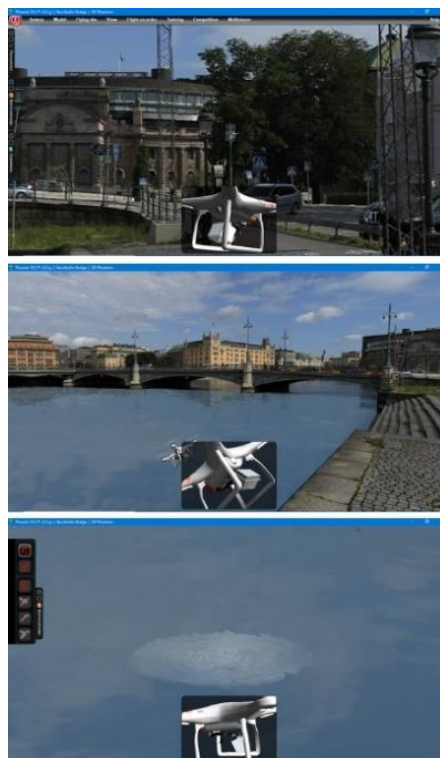


Figura 0.14 Simulación en medio de la ciudad [Los autores]

CAPÍTULO 5

RESULTADOS OBTENIDOS (VALIDACIÓN DEL FUNCIONAMIENTO)

5.1. PRUEBAS PRELIMINARES

Para esta etapa se deben remover las hélices de la aeronave por motivos de seguridad y protección, seguido de ello se conecta la batería al módulo de alimentación para así alimentar el sistema autopiloto y sus demás componentes.

5.1.1. Transmisión de datos

En primera estancia, se ejecuta el programa QGroundControl, y a continuación se conecta mediante cable micro USB el segundo módulo de telemetría para así entablar comunicación con la aeronave. Tras ello, en la parte superior de la pantalla se pueden observar diferentes iconos de estado como lo son:

Mensajes del vehículo: En esta sección se muestra una lista de mensajes correspondientes al estado del vehículo y en caso de que este se encuentre en estado crítico el icono cambia por un símbolo de emergencia. A demás de ello, el software avisa al usuario brindando la información por audio.



Estado del GPS: Indica el número de satélites actuales disponibles.



RC RSSI: Indica la intensidad de la señal del radio control.



Telemetry RSSI: Indica la Intensidad de la señal del módulo de telemetría.



Batería: Indica el porcentaje de batería restante.



Modo de vuelo: Indica el modo de vuelo actual de la aeronave.

5.1.2. Vista de vuelo

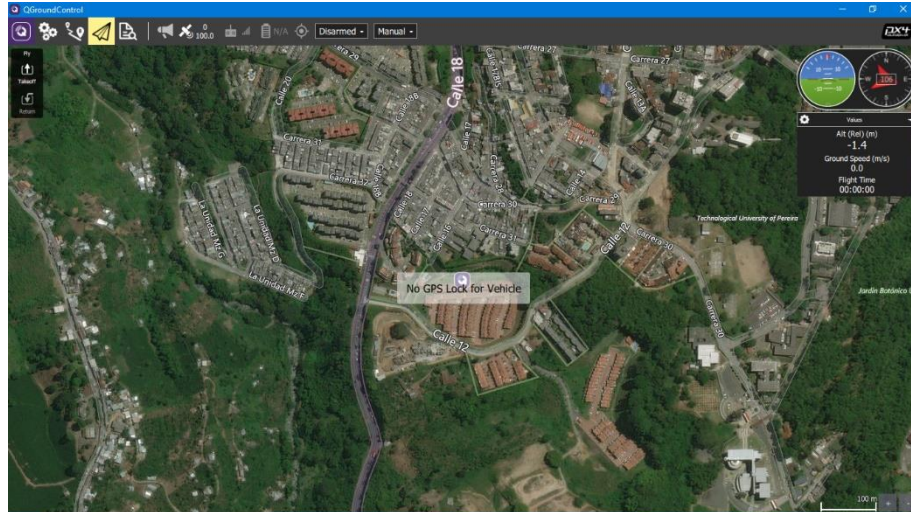


Figura 0.1 Ventana de vista de vuelo de QGroundControl [Los autores]

Esta pestaña se utiliza para controlar y supervisar el vehículo cuando está volando. Entre sus principales usos se encuentran:

- Ejecutar una comprobación automática del estado de la aeronave previa al vuelo.
- Cambiar entre la visualización del mapa o vista de video si está configurada.
- Mostrar información sobre la telemetría y otra información disponible.

En la pestaña “Values” se muestra la información de telemetría por defecto, correspondiente a la altitud con respecto al punto de partida y la velocidad de la tierra. Adicionalmente se pueden añadir más parámetros, al presionar el engranaje de la parte superior izquierda y eligiendo su formato de visualización. Lo anterior es mostrado en la Figura 0.2:

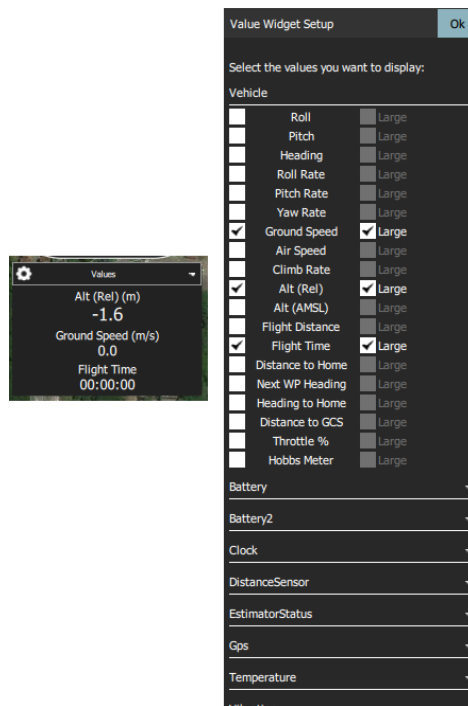


Figura 0.2 Pestaña de Valores y parámetros de medición [Los autores]

En la parte superior derecha se muestra la actitud de la aeronave e información sobre la brújula de esta. Estas se muestran en la Figura 0.3:



Figura 0.3 Visualización de la actitud y la brújula [Los autores]

5.1.3. Testeo de los motores

En primera media, se presiona el interruptor de seguridad hasta que el sistema autopiloto emita una secuencia de sonido indicado que la aeronave está armada. Dentro de la pestaña de “Vehicle Setup” se va a la opción de “Motors”, donde se desplazará el slider mostrado en pantalla para habilitar los motores, a continuación se procede a probar cada uno de forma individual con su respectivo slider y finalmente de forma conjunta. Tras culminar el proceso, se corre hacia la izquierda el slider correspondiente para deshabilitar los motores y se procede a desarmar la aeronave presionando el interruptor de seguridad. El procedimiento realizado se muestra en la Figura 0.4:



Figura 0.4 Testeo de los motores [Los autores]

5.2. PRUEBA DE LA AERONAVE

Para esta etapa se deben ubicar las hélices en los motores correspondientes. Por motivos de seguridad y velando por la integridad de quien pilotea la aeronave, las pruebas descritas a continuación se deben llevar a cabo en un campo abierto.

5.2.1. Posicionamiento y armado de la aeronave

En primera medida, se posiciona la aeronave de forma frontal, es decir, que los brazos de color rojo del chasis apunten hacia adelante. A continuación, se presiona el interruptor de seguridad y se espera a que el sistema autopiloto indique que la aeronave está armada y lista para volar; seguido de esto, quien maneja la aeronave debe posicionarse a una distancia de entre 2.5 y 4 metros, de forma que pueda observar la parte trasera de la aeronave, tal como se muestra en la Figura 0.5:

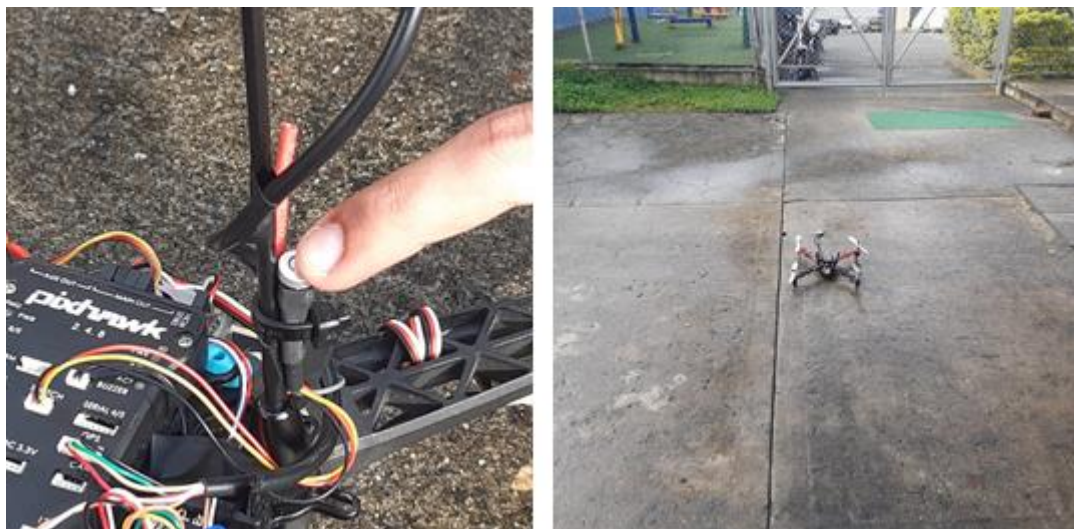


Figura 0.5 Posicionamiento y armado de la aeronave [Los autores]

Adicional al accionamiento del interruptor de seguridad, se debe seleccionar el modo de vuelo configurado previamente, para este caso el modo manual, y para ello, se pone cambia la posición del switch superior derecho de 0 a 1. A continuación, se debe ubicar la palanca de “Throttle” (palanca izquierda), en el límite inferior derecho, con ello, los motores de la aeronave se encenderán a una velocidad baja lo que indica que ya se puede manejar la aeronave por medio del control remoto. El proceso se muestra en la Figura 0.6:

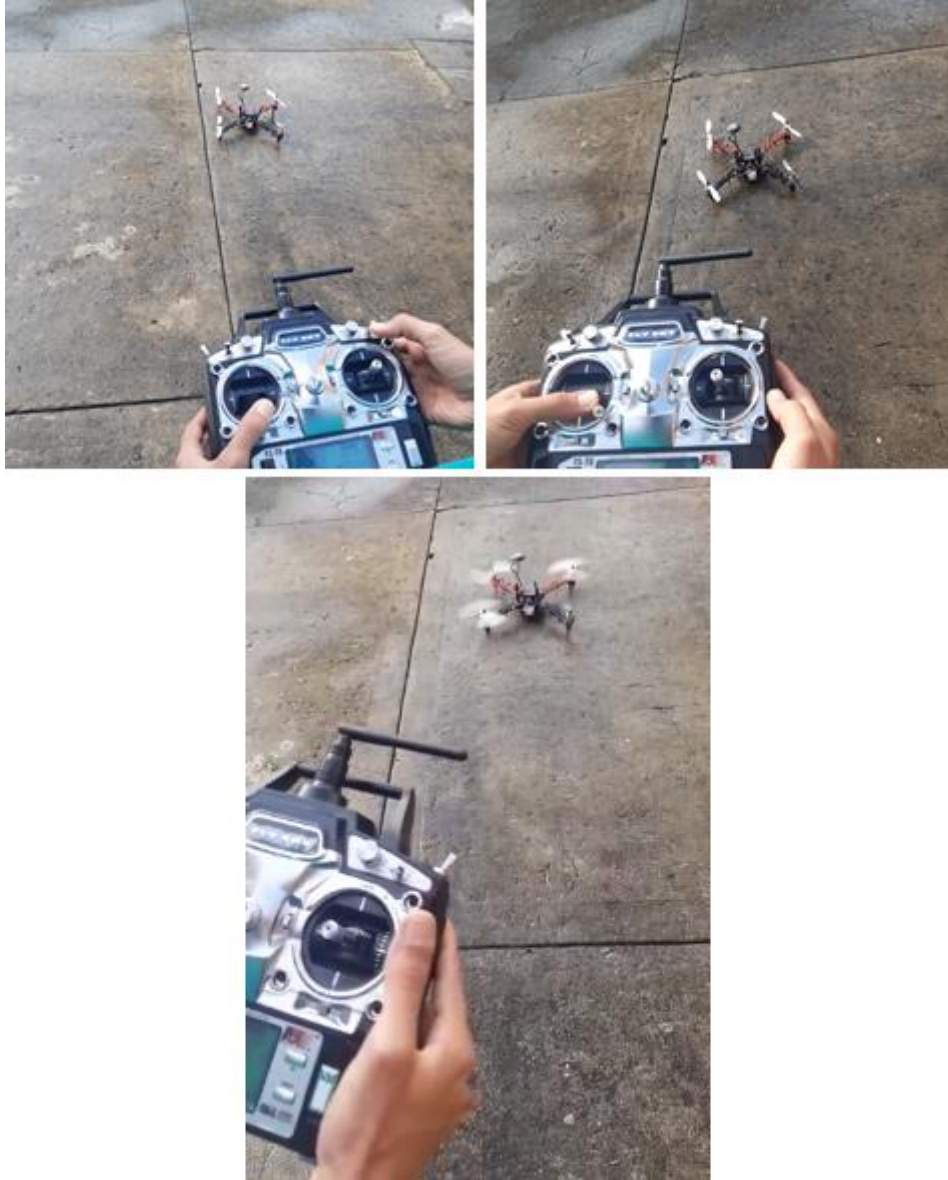


Figura 0.6 Preparación y encendido de la aeronave [Los autores]

6. PRESUPUESTO

COSTO DE EJECUCIÓN DE LAS ACTIVIDADES					
ID ACTIVIDAD	DESCRIPCIÓN	RUBRO	TIEMPO DE EJECUCIÓN	VALOR UNITARIO	VALOR TOTAL
1.1	Búsqueda de información por medio de la web	Acceso a la información	6 horas	3700	22200
1.2	Obtención, análisis y clasificación de la información	Costo personal técnico no especializado	4 horas	3700	14800
2.1	Delimitación de la propuesta	Costo personal técnico no especializado	3 horas	3700	11100
2.2	Entrega y revisión del primer borrador	Costo personal especializado	2 horas	60000	120000
3.1	Definición de conceptos básicos y ejemplificación	Costo personal técnico no especializado	4 horas	3700	14800
3.2	Clasificación de sistemas autopilotos existentes en el mercado y de los diferentes softwares de control	Costo personal técnico no especializado	4 horas	3700	14800
3.3	Caracterización de software de programación, control y simulación	Costo personal técnico no especializado	2 horas	3700	7400
3.4	Entrega y revisión del segundo borrador	Costo personal especializado	2 horas	60000	120000
4.1	Selección del sistema autopiloto a implementar	Costo personal especializado	1 hora	3700	3700
4.2	Selección del software de programación y control	Costo personal técnico no especializado	3 horas	3700	11100

4.3	Carga del firmware al sistema autopiloto, conexiones preliminares y calibración de sensores	Costo personal técnico no especializado	4 horas	3700	14800
5.1	Principios, ensamble y acoplamiento de componentes a la estructura principal	Costo personal técnico no especializado	8 horas	3700	29600
5.2	Entrega y revisión del tercer borrador	Costo personal especializado	2 horas	60000	120000
6.1	Selección del entorno de simulación	Costo personal técnico no especializado	1 hora	3700	3700
6.2	Realización de simulaciones más ajustes	Costo personal técnico no especializado	12 horas	3700	44400
6.3	Validación de funcionamiento en plataforma aérea de vuelo vertical	Costo personal técnico no especializado	4 horas	3700	14800
6.4	Entrega y revisión final	Costo personal especializado	2 horas	60000	120000
TOTAL					687200

Tabla 6.1 Coste de ejecución de las actividades

RECURSOS MATERIALES	CANTIDAD	RESPONSABLE	VALOR UNITARIO	VALOR TOTAL
Motores	4	UTP (laboratorio de aeronáutica)	49000	196000
Hélice	4	UTP (laboratorio de aeronáutica)	9000	36000
Drivers esc 30A	4	UTP (laboratorio de aeronáutica)	34000	136000
Batería	1	UTP (laboratorio de aeronáutica)	23000	23000
Cargador	1	Grupo de trabajo	30000	30000
Control para pruebas	1	UTP (laboratorio de aeronáutica)	375000	375000

Chasis DJI F450	1	UTP (laboratorio de aeronáutica)	170000	170000
Kit Pixhawk	1	Grupo de trabajo	450000	450000
Conectores de Batería	2	Grupo de trabajo	1900	3800
Computador	1	Grupo de Trabajo	2500000	1000000
Mano de Obra	2	Grupo de Trabajo	600000	600000
Total	22		Total	3019800

Tabla 6.2 Recursos materiales

7. CONCLUSIONES

Los resultados obtenidos en el trabajo de grado cumplen el objetivo general de desarrollar un sistema programable micro controlado para control de una aeronave de despegue vertical y vuelo horizontal, identificando una cantidad de inconvenientes al realizar las pruebas que llevaron a afianzar los conocimientos, al final validando con éxito el vuelo vertical.

Como consecuencia de la investigación de identificar los diferentes sistemas autopilotos ya existentes en el mercado y sus características de funcionamiento básicas, optamos por usar PIXHAWK por su economía, calidad de control y es didáctico para programar.

La programación del autopiloto instalando el programa firmware y usando el programa QGroundControl llegando a excelentes resultados en las pruebas de vuelo.

Concluimos que fue muy didactico y facil la configuracion de sensores, y de la aeronave en general haciendo que el sistema software y hardware, comportándose de manera adecuada durante el vuelo.

Las simulaciones realizadas mediante un software especializado se hicieron controlada mediante un celular ya que los otros softwares presentaron dificultades en la instalación y fue dispendioso su acople, dando excelentes resultados.

Cualquier persona que quiera realizar el montaje de este desarrollo microcontrolado PIXHAWK adaptado en este tipo de drones, podrá encontrar en este trabajo toda la información necesaria para solucionar sus dudas.

8. RECOMENDACIONES

Cuando se está iniciando el proceso de controlar una aeronave de piloto autónomo, se recomienda usar Pixhawk, ya que es un sistema económico, además presta la alternativa de realizar vuelos alternos a los de un dron convencional, se recomienda por su software es didáctico.

Se recomienda usar este documento ya que está plasmado el paso a paso de la calibración y programación de un Pixhawk

Por motivos de seguridad antes de realizar cualquier proceso de calibración o prueba estacionaria con el sistema PIXHWAK se recomienda hacerlo en un área cerrada, sin las hélices para evitar cualquier accidente.

A pesar de que el PIXHWAK tiene en su telemetría alcances muy amplios, se pueden mejorar mediante el uso de una antena adicional o el cambio por transmisores más potentes o reduciendo la misma tasa de datos enviados.

El uso de un segundo módulo de GPS mejora el funcionamiento del sistema de control de vuelo automático en cuanto a redundancia de la señal de GPS al ser un sistema tan dependiente de este mismo módulo, siendo una opción que se puede explorar a futuro.

Se recomienda no usar tornillería improvisada o elementos adicionales sin previos diseños, ya que pueden afectar el vuelo o dañar algunas como motores hélices entre otros.

9. BIBLIOGRAFÍA

- [1] E.E.G, «MuyCanal,» 13 Diciembre 2019. [En línea]. Available: <https://www.muycanal.com/2019/12/13/ventas-drones-empresariales>.
- [2] «Héliceo: Fusion, Drone avión de despegue vertical,» [En línea]. Available: <http://www.heliceo.com/es/produits-pour-geometres/fusion-dron-avion-vtol/>.
- [3] «Wikipedia la enciclopedia libre: Vehículo aéreo no tripulado,» [En línea]. Available: https://es.wikipedia.org/wiki/Veh%C3%ADculo_a%C3%A9reo_no_tripulado#Clasificaci%C3%B3n_de_los_drones.
- [4] «Mundo Sputnik News: UAV de reconocimiento Korsar de origen ruso,» [En línea]. Available: <https://mundo.sputniknews.com/defensa/201805111078600648-defensa-ffaa-drones-moscu-washington/>.
- [5] «Wikipedia la enciclopedia libre: General Atomics MQ-9 Reaper,» [En línea]. Available: https://es.wikipedia.org/wiki/General_Atomics_MQ-9_Reaper.
- [6] «DJI: DJI Phantom 4 Pro V2.0,» [En línea]. Available: https://www.dji.com/phantom-4-pro-v2?site=brandsite&from=landing_page.
- [7] «RACER.LT: INAV FLIGHT F4 Flight Controller,» [En línea]. Available: <https://www.racer.lt/item/inav-flight-f4-flight-controller-built-in-osd-battery-voltage-current-monitor-for-fpv-rc-airplane-12979.html>.
- [8] «FPVManía: Que es naze32,» 7 Febrero 2015. [En línea]. Available: <https://fpvmania.es/tutorial-naze32-parte-1-que-es-y-como-se-monta-una-naze32/>.
- [9] «Aliexpress: Openpilot-CONTROLADOR DE VUELO CC3D EVO,» [En línea]. Available: <https://es.aliexpress.com/item/4000288855992.html>.
- [10] «LibrePilot,» [En línea]. Available: <https://www.librepilot.org/site/index.html>.
- [1] M. Hubáček, «@librepilot,» 23 Septiembre 2016. [En línea]. Available: <https://twitter.com/librepilot>.
- [1] «DJI: NAZA DJI,» [En línea]. Available: <https://www.dji.com/naza-m-v2>.
- [1] «ARDUPILOT: Versatile, Trusted, Open,» [En línea]. Available: <https://ardupilot.org/ardupilot/>.

- [1] «Aliexpress: Placa de control de vuelo APM 2.8,» [En línea]. Available:
4] <https://es.aliexpress.com/item/32650138120.html?src=bing&albslr=112899629&isdl=y>.
- [1] «BetaFlight,» [En línea]. Available: <https://betaflight.com/>.
5]
- [1] «READYTOSTKY: SP Racing F3 Flight Controller ACRO Version,» [En
6] línea]. Available: http://www.readytsky.com/e_products/show/?799-SP-Racing-F3-Flight-Controller-ACRO-Version-799.html.
- [1] «dRonin: Advanced Flight Control,» [En línea]. Available: <https://dronin.org/>.
7]
- [1] «hobbyking.com: MultiWii 328P regulador de vuelo,» [En línea]. Available:
8] https://hobbyking.com/es_es/multiwii-328p-flight-controller-w-ftdi-dsm2-comp-port.html?___store=es_es.
- [1] «Ardupilot: TauLabs Sparky2,» [En línea]. Available:
9] <https://ardupilot.org/copter/docs/common-taulabs-sparky2.html>.
- [2] «Ardupilot: Pixhawk Overview,» [En línea]. Available:
0] <https://ardupilot.org/copter/docs/common-pixhawk-overview.html>.
- [2] A. D. Team, «Ardupilot: APM Planner 2 Home,» 2020. [En línea]. Available:
1] <https://ardupilot.org/planner2/>.
- [2] «Dronecode: QGroundControl User Guide,» [En línea]. Available:
2] <https://docs.qgroundcontrol.com/master/en/index.html>.
- [2] A. D. Team, «Ardupilot: Mission Planner Home,» 2020. [En línea].
3] Available: <https://ardupilot.org/planner/index.html#home>.
- [2] «Dronecode: jMAVSim with STL,» [En línea]. Available:
4] <https://dev.px4.io/v1.9.0/en/simulation/jmavsim.html>.
- [2] «Dronecode: FlightGear Simulation,» [En línea]. Available:
5] <https://dev.px4.io/master/en/simulation/flightgear.html>.
- [2] «Dronecode: Gazebo Simulation,» [En línea]. Available:
6] <https://dev.px4.io/v1.9.0/en/simulation/gazebo.html>.
- [2] «Dronecode: AirSim Simulation,» [En línea]. Available:
7] <https://dev.px4.io/master/en/simulation/airsim.html>.
- [2] «hobbyplay: Simulador Phoenix V5,» [En línea]. Available:
8] https://www.hobbyplay.net/simulacion/simulador_phoenix_rc.
- [2] «DroidJoy Virtual PC Gamepad,» [En línea]. Available:
9] <https://grill2010.github.io/droidJoy.html>.

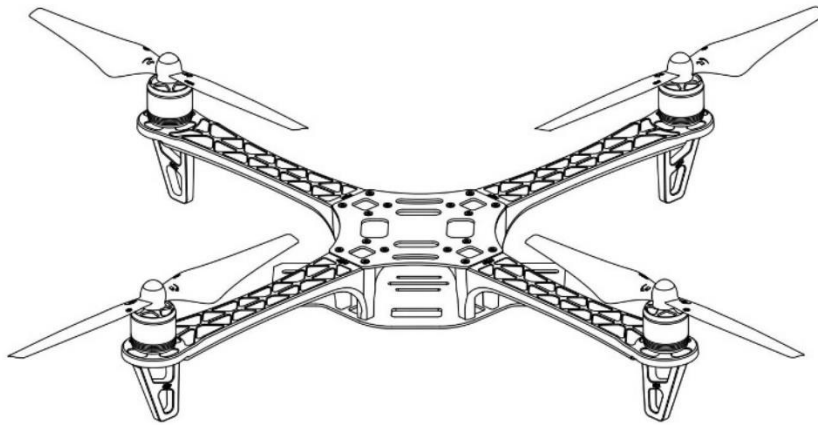
- [3] «Geeetech: 3DR Power module users manual,» [En línea]. Available:
0] <https://www.geeetech.com/Documents/3DR%20Power%20Module.pdf>.
- [3] «U-BLOX: NEO-M8,» 30 Abril 2020. [En línea]. Available: <https://www.u-blox.com/en/docs/UBX-15031086>.
- [3] «Amazon: Telemetría inalámbrica 1000 mW,» [En línea]. Available:
2] <https://www.amazon.com.mx/Hobbypower-Telemetr%C3%ADa-inal%C3%A1mblica-APM2-8-Pixhawk/dp/B07Q137YSP>.
- [3] «Ardupilot: Codificador PPM,» [En línea]. Available:
3] <https://ardupilot.org/copter/docs/common-ppm-encoder.html>.
- [3] «REMOTE CONTROL DRONES: FlySky FS-T6,» [En línea]. Available:
4] <https://getrcdrones.com/tag/flysky/>.

10. ANEXOS

FlameWheel 450

User Manual V2.2

2015.05 Revision



Disclaimer

Read this disclaimer carefully before using FlameWheel 450. By using this product, you hereby agree to this disclaimer and signify that you have read them fully. FlameWheel 450 is an excellent multi-rotor. With a good autopilot, it will even offer tremendous flight features for low altitude flight working in restricted space. Despite the controller of autopilot operates in the safest manner when the main power battery is connected, we strongly recommend customers to remove all propellers, use power supply from R/C system or flight pack battery, and keep children away during system calibration and parameter setup. Please respect the AMA's National Model Aircraft Safety Code. Please install and use this product in strict accordance with the User Manual. SZ DJI Technology Co., Ltd. and its affiliated companies assume no liability for damage(s) or injuries incurred directly or indirectly from using, installing or refitting this product improperly, including but not limited to using non-designated accessories.

This device complies with part 15 of the FCC Rules.

DJI FLAME WHEEL is registered trademark of DJI Innovations. Names of product, brand, etc., appearing in this manual are trademarks or registered trademarks of their respective owner companies. This product and manual are copyrighted by DJI Innovations with all rights reserved. No part of this product or manual shall be reproduced in any form without the prior written consent or authorization of DJI Innovations. No patent liability is assumed with respect to the use of the product or information contained herein.

F450 Profile

FlameWheel 450 (F450) is a multi-rotor designed for all pilots for fun or AP. With DJI WKM or NAZA autopilot system, it can achieve hovering, cruising, even rolling and other flight elements. It can be applied for entertainment, aerial photography, FPV and other aero-modeling activities.

Product Usage Cautions

When flying, the fast rotating propellers of FlameWheel 450 will cause serious damage and injury. Therefore, please fly with a high safety consciousness.

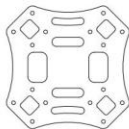
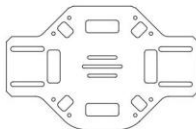






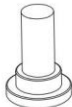


1. Keep flying multi- rotor away from objects, such as obstacles, human beings, high- voltage lines and so on.
2. Do not get close to or even touch the working motors and propellers, which will cause serious injury.
3. Do not over load the multi- rotor.
4. Check that the propellers and the motors are installed correctly and firmly before flight.
5. Check whether all parts of multi- rotor are in good condition before flight. Do not fly with old or broken parts.
6. Use DJI parts as much as possible.

If you have any problem you cannot solve during installation, please contact our customer service.

Contents

Disclaimer	2
F450 Profile	2
Product Usage Cautions	2
Contents	3
In the Box	4
Tools Needed	5
Preparing for Mounting	5
Motor Mounting Description	5
Propeller Mounting Description	6
ESC and Motor Wiring Description	6
Assembly	7
ESC Sound Description	8
Specifications	8

In the Box

Top Board 450FBT × 1	Bottom Board 450FBB × 1
	
Frame Arms 450FAC × 2	Frame Arms 450FAW × 2
	
Motors CW × 2	Motors CCW × 2
	
Propeller Pairs × 3	ESC × 4
	
Screws 450-M3 × 8 × 16	Screws 450-M2.5 × 6 × 24
	
Propeller Removal Clamp × 1	Magic Strap 450MSX × 1
	Battery Band 450BBX × 11 Power Line Pair 450PLP × 1

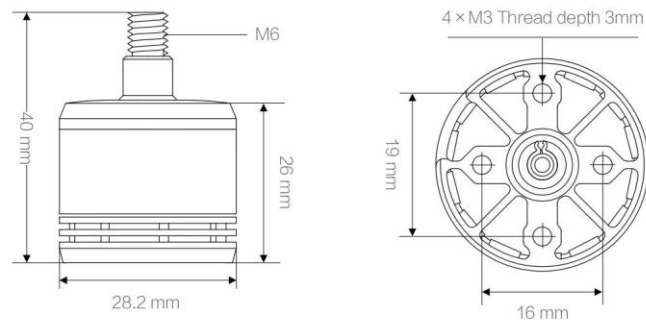
Tools Needed

Tools	Use
2.0mm Hex Wrench	For frame and motors installation.
Screw Glue	For fastening screws.
Nylon Cable Tie, Scissors, Diagonal Cutting Pliers	For binding devices and wires.
Foam Double Sided Adhesive Tape	For fixing receiver, controller and other modules.
Soldering-iron & wires	For connecting ESCs' power cables to bottom board.

Preparing for Mounting

Motor Mounting Description

The size of the assembly hole is shown below.



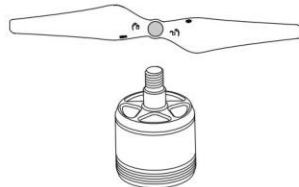
-
- ⚠
- The screw size is M3. Mount the motors using appropriate screws.
 - Note the thread depth and the size of the screws. Using screws that are too long or too large may damage the motor.
-

Propeller Mounting Description

Attach the propeller with a silver nut onto the counter clockwise motor.


Attach the propeller with a black nut onto the clockwise motor (which has an indent on the screw).


Silver nut (9450)



Black nut (9450 cw)

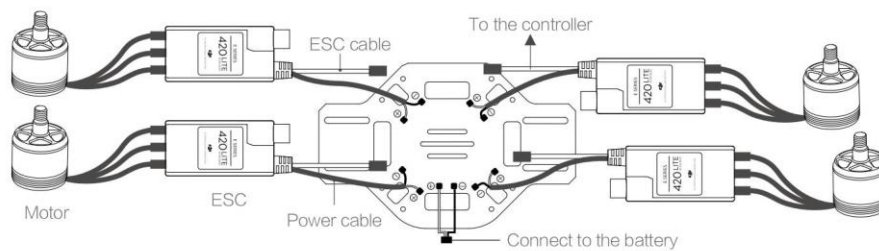


 Lock: Tighten the propeller by rotating it in this direction.

 Unlock: Remove the propeller by rotating it in this direction.

ESC and Motor Wiring Description

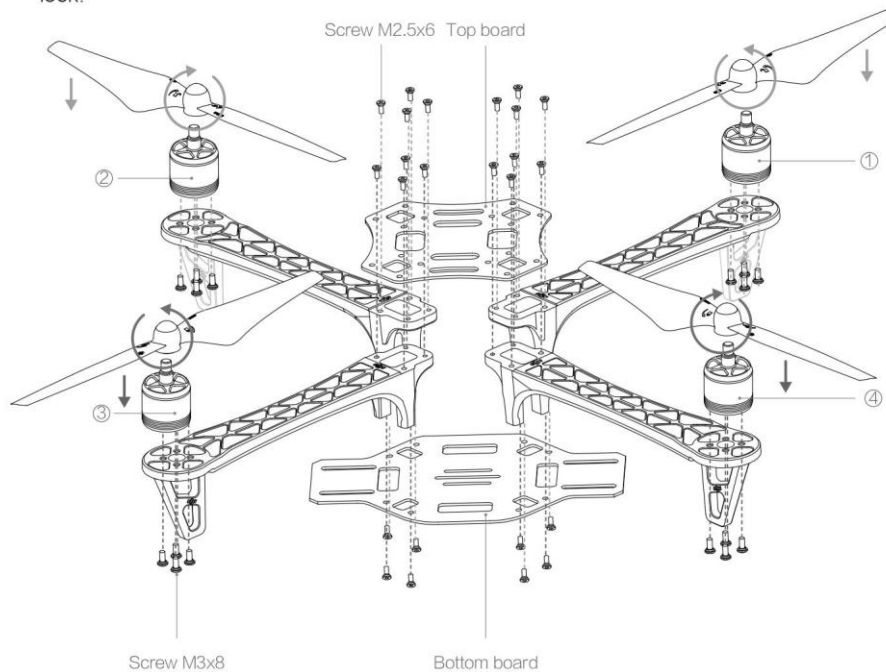
1. Please solder each ESC to the power pads on the bottom board as shown in the figure below. Make sure that the solder is firmly attached on the power pads and that there is no possibility for a short circuit. The power cable is a coaxial power cable. Do not damage the protector on the red cable to avoid short circuit. The red core of the coaxial power cable is positive, and the shielding layer of the coaxial power cable is negative. Make sure to solder them correctly at the proper length (the wires should be just long enough to reach the soldering point, face outwards, and should not be bunched up in front of the soldering point).
2. Connect the signal cable to your controller. The orange wire of the signal cable is for the control signal; the brown wire of the signal cable is for the GND.
3. Connect the motor to the ESC. Test the motor and make sure the rotation direction of each motor is correct. If not, switch the position of any two cables that are connecting the motor to the ESC to change the rotation direction.



- ⚠
- Please solder ESCs to power pads on bottom board as the figure shows.
 - Use any insulating method at all soldering spots. Make sure there is no short or open circuit.
 - Make sure the side of power pads is upward.

Assembly

1. Install the bottom board. Note to install the screws by appropriate force to prevent breaking threads. Use adequate screw glue for installing screws.
2. Install your flight control system and electronic system.
3. Install ESCs and motors. Make sure the rotation direction of each motor is the same as the way in the figure shows. If not, switch any of two wire connections of the incorrect motor to change its rotation direction.
4. Tidy all cables. Make sure all cables are not be cut by the frame boards and propellers. Smooth out the boards edge if necessary.
5. Install the top board.
6. Install the propellers. Please install the propellers after the flight control system configuration procedure. Make sure the rotation direction of propellers are the same as the figure shows. Tighten the propeller by rotating it in lock direction. Do not use any thread lock.



- ⚠ Note that arms ①② point to the nose of aircraft and arms ③④ point to the tail of aircraft. It is recommended to choose the same color arms for the same direction. See from top, motors on arms ①③ rotate counter-clock wise and motors on arms ②④ rotate clock wise.

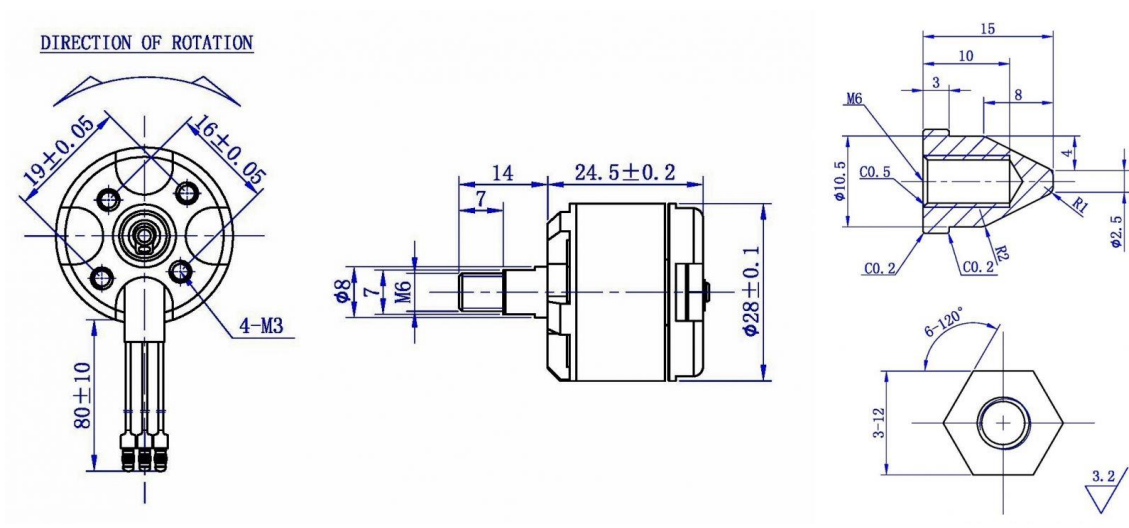
ESC Sound Description

Normal	Description
♪ 1234	Ready
Abnormal	Description
BBBBBB...	No signal input, or throttle stick is not in the bottom position

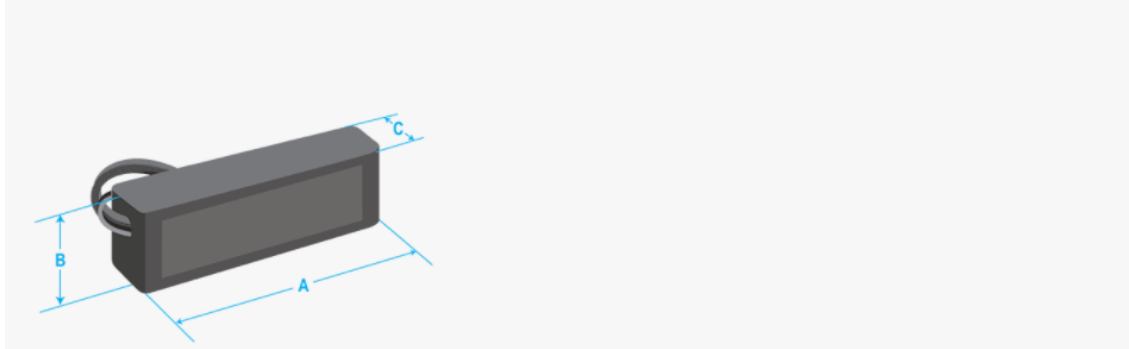
- 💡 You can instantly tell what the unit's status is by listening to the sounds emitted from the ESC.

Specifications

Frame	
Diagonal Wheelbase	450 mm
Frame Weight	282 g
Takeoff Weight	800 g ~ 1600 g
ESC	
Max Allowable Voltage	17.4 V
Max Allowable Current (Persistent)	20 A
Max Allowable Peak Current (3 seconds)	30 A
PWM Input Signal Level	3.3 V / 5 V Compatible
Signal Frequency	30 Hz ~ 450 Hz
Battery	3 S ~ 4S LiPo
Weight (without cable)	12.5 g
Weight (with cable)	27 g
Motor	
Stator size	23 × 12 mm
KV	960 rpm/V
Weight	57 g
Propeller	
Diameter / Thread Pitch	24 × 12.7 cm (9.4 × 5.0inch)
Weight (Single)	13 g



Capacity (mAh)	2200.00	Discharge(c)	25.00
Max Charge Rate(C)	5.00	Length-A(mm)	107.00
Height-B(mm)	34.00	Width-C(mm)	24.00





NEO-M8

u-blox M8 concurrent GNSS modules

Data sheet



Abstract

Technical data sheet describing the NEO-M8 module family which provides concurrent reception of up to 3 GNSS (GPS, Galileo, GLONASS, BeiDou). NEO-M8 is backward compatible with NEO-7, NEO-6 and NEO-5 families.

www.u-blox.com

UBX-15031086 - R07



Document information

Title	NEO-M8	
Subtitle	u-blox M8 concurrent GNSS modules	
Document type	Data sheet	
Document number	UBX-15031086	
Revision and date	R07	30-Apr-2020
Document status	Production Information	

Product status	Corresponding content status	
In Development / Prototype	Objective Specification	Target values. Revised and supplementary data will be published later.
Engineering Sample	Advance Information	Data based on early testing. Revised and supplementary data will be published later.
Initial Production	Early Production Information	Data from product verification. Revised and supplementary data may be published later.
Mass Production / End of Life	Production Information	Document contains the final product specification.

This document applies to the following products:

Product name	Type number	Firmware version	PCN reference
NEO-M8N	NEO-M8N-0-11	Flash FW SPG 3.01	UBX-20013367
NEO-M8Q	NEO-M8Q-0-11	ROM SPG 3.01	UBX-20013367
NEO-M8M	NEO-M8M-0-10	ROM SPG 3.01	UBX-16013121

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.

The information contained herein is provided "as is" and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

Copyright © u-blox AG.

Contents

Document information	2
Contents	3
1 Functional description	5
1.1 Overview	5
1.2 Product features	5
1.3 Performance.....	6
1.4 Block diagram	7
1.5 Supported GNSS constellations	7
1.5.1 GPS	7
1.5.2 GLONASS	8
1.5.3 BeiDou	8
1.5.4 Galileo	8
1.6 Assisted GNSS (A-GNSS).....	8
1.6.1 AssistNow™ Online.....	8
1.6.2 AssistNow™ Offline.....	9
1.6.3 AssistNow™ Autonomous	9
1.7 Augmentation systems.....	9
1.7.1 Satellite-based augmentation system (SBAS)	9
1.7.2 QZSS	9
1.7.3 IMES	10
1.7.4 Differential GPS (D-GPS).....	10
1.8 Broadcast navigation data and satellite signal measurements	10
1.9 Odometer.....	10
1.10 Data logging (NEO-M8N).....	10
1.11 Geofencing.....	11
1.12 Message integrity protection	11
1.13 Spoofing detection	11
1.14 EXTINT: External interrupt.....	11
1.14.1 Pin control	11
1.14.2 Aiding	11
1.15 TIMEPULSE.....	12
1.16 Protocols and interfaces	12
1.17 Interfaces.....	12
1.17.1 UART	12
1.17.2 USB	12
1.17.3 SPI.....	12
1.17.4 Display data channel (DDC)	13
1.18 Clock generation.....	13
1.18.1 Oscillators	13
1.18.2 Real-time clock (RTC)	13

1.19 Power management	13
1.19.1 DC-DC converter	13
1.19.2 Power mode setup	13
1.19.3 Continuous mode.....	14
1.19.4 Power save mode	14
1.20 Antenna.....	14
1.21 LNA_EN.....	14
2 Pin definition	15
2.1 Pin assignment.....	15
2.2 Pin name changes.....	16
3 Configuration management.....	17
3.1 Interface selection (D_SEL)	17
4 Electrical specification	18
4.1 Absolute maximum rating.....	18
4.2 Operating conditions.....	18
4.3 Indicative current requirements	19
4.4 SPI timing diagrams.....	20
4.4.1 Timing recommendations.....	20
4.5 DDC timing diagrams	20
5 Mechanical specifications	21
6 Reliability tests and approvals.....	22
6.1 Reliability tests.....	22
6.2 Approvals.....	22
7 Product handling and soldering	23
7.1 Packaging	23
7.1.1 Reels	23
7.1.2 Tapes.....	23
7.2 Shipment, storage and handling	24
7.2.1 Moisture sensitivity levels	24
7.2.2 Reflow soldering.....	24
7.2.3 ESD handling precautions	24
8 Default messages	25
9 Labeling and ordering information	26
9.1 Product labeling.....	26
9.2 Explanation of codes	26
9.3 Ordering codes.....	26
Appendix	27
A Glossary	27
Related documents	28
Revision history	28
Contact.....	29

1 Functional description

1.1 Overview

The NEO-M8 series of concurrent GNSS modules is built on the high-performing u-blox M8 GNSS engine in the industry proven NEO form factor.

The NEO-M8 series utilizes concurrent reception of up to three GNSS systems (GPS/Galileo together with BeiDou or GLONASS), recognizes multiple constellations simultaneously and provides outstanding positioning accuracy in scenarios where urban canyon or weak signals are involved. For even better and faster positioning improvement, the NEO-M8 series supports augmentation of QZSS, GAGAN and IMES together with WAAS, EGNOS, and MSAS. The NEO-M8 series also supports message integrity protection, geofencing, and spoofing detection with configurable interface settings to easily fit to customer applications. The NEO form factor allows easy migration from previous NEO generations.

The NEO-M8M is optimized for cost-sensitive applications, while NEO-M8N/M8Q provides best performance and easier RF integration. The NEO-M8N offers high performance also at low power consumption levels. The future-proof NEO-M8N includes an internal flash that allows future firmware updates. This makes NEO-M8N perfectly suited to industrial and automotive applications.

The DDC (I2C-compliant) interface provides connectivity and enables synergies with most u-blox cellular modules. For RF optimization the NEO-M8N/Q features an additional front-end LNA for easier antenna integration and a front-end SAW filter for increased jamming immunity.

u-blox M8 modules use GNSS chips qualified according to AEC-Q100, are manufactured in ISO/TS 16949 certified sites, and fully tested on a system level. Qualification tests are performed as stipulated in the ISO16750 standard: "Road vehicles – Environmental conditions and testing for electrical and electronic equipment".

The u-blox NEO-M8 modules can also benefit from the u-blox AssistNow assistance service. The Online service provides GNSS broadcast parameters, for example, ephemeris, almanac plus time or rough position to reduce the receiver's time to first fix significantly and to improve acquisition sensitivity. The extended validity of AssistNow Offline data (up to 35 days) and AssistNow Autonomous data (up to 6 days) provide faster acquisition after a long off time.

See section 1.6 for more information concerning the NEO-M8 related AssistNow Assistance.

1.2 Product features

Model	Category	GNSS	Supply	Interfaces	Features	Grade
	Standard Precision GNSS High Precision GNSS Dead Reckoning Timing	GPS / QZSS GLONASS Galileo BeiDou Number of concurrent GNSS	1.65 V – 3.6 V 2.7 V – 3.6 V	UART USB SPI DDC (I ² C compliant)	Programmable (flash) Data logging Additional SAW Additional LNA RTC crystal Oscillator Built-in antenna Built-in antenna supply and supervisor Timepulse	Standard Professional Automotive
NEO-M8N	•	• • • • • 3	•	• • • • •	• • • • • T	1 •
NEO-M8Q	•	• • • • • 3	•	• • • • •	• • • • T	1 •
NEO-M8M	•	• • • • • 3	•	• • • • •	• C	1 •

C = Crystal / T = TCXO

1.3 Performance

Parameter	Specification					
Receiver type	72-channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C					
Accuracy of time pulse signal	RMS	30 ns				
	99%	60 ns				
Frequency of time pulse signal	0.25 Hz...10 MHz (configurable)					
Operational limits ¹	Dynamics	≤ 4 g				
	Altitude	50,000 m				
	Velocity	500 m/s				
Velocity accuracy ²	0.05 m/s					
Heading accuracy ²	0.3 degrees					
GNSS	GPS & GLONASS		GPS	GLONASS	BeiDou	Galileo
Horizontal position accuracy ³	2.5 m		2.5 m	4 m	3 m	TBC ⁴
NEO-M8N/Q						
Max navigation update rate	NEO-M8N	5 Hz	10 Hz	10 Hz	10 Hz	10 Hz
	NEO-M8Q	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix ⁵	Cold start	26 s	29 s	30 s	34 s	45 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts ⁶	2 s	2 s	2 s	3 s	7 s
Sensitivity ⁷	Tracking & Navigation	-167 dBm	-166 dBm	-166 dBm	-160 dBm	-159 dBm
	Reacquisition	-160 dBm	-160 dBm	-156 dBm	-157 dBm	-153 dBm
	Cold start	-148 dBm	-148 dBm	-145 dBm	-143 dBm	-138 dBm
	Hot start	-157 dBm	-157 dBm	-156 dBm	-155 dBm	-151 dBm
NEO-M8M						
Max navigation update rate		10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix ⁵	Cold start	26 s	30 s	33 s	39 s	57 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts ⁶	3 s	3 s	3 s	7 s	7 s
Sensitivity ⁷	Tracking & Navigation	-164 dBm	-164 dBm	-163 dBm	-160 dBm	-154 dBm
	Reacquisition	-160 dBm	-159 dBm	-156 dBm	-155 dBm	-152dBm
	Cold start	-148 dBm	-147 dBm	-145 dBm	-143 dBm	-133 dBm
	Hot start	-157 dBm	-156 dBm	-155 dBm	-155 dBm	-151 dBm

Table 1: NEO-M8 performance in different GNSS modes (default: concurrent reception of GPS and GLONASS including QZSS, SBAS)

¹ Assuming Airborne < 4 g platform

² 50% at 30 m/s

³ CEP, 50%, 24 hours static, -130 dBm, > 6 SVs

⁴ To be confirmed when Galileo reaches full operational capability

⁵ All satellites at -130 dBm, except Galileo at -127 dBm

⁶ Dependent on aiding data connection speed and latency

⁷ Demonstrated with a good external LNA

1.4 Block diagram

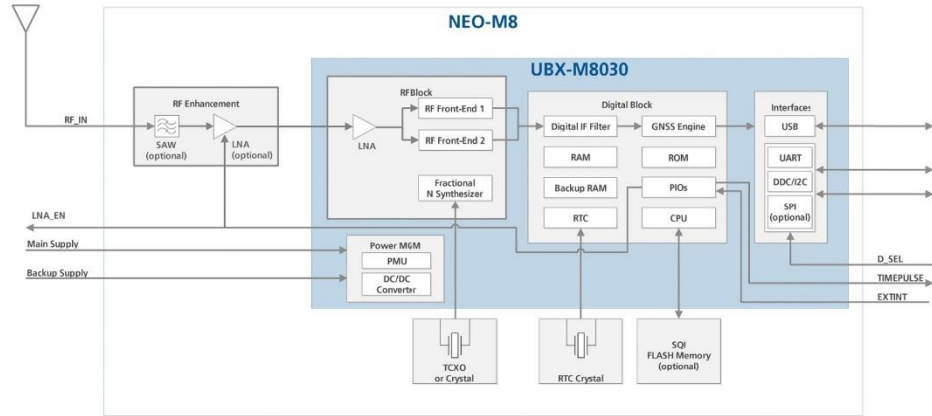


Figure 1: NEO-M8 block diagram

1.5 Supported GNSS constellations

The NEO-M8 GNSS modules are concurrent GNSS receivers which can receive and track multiple GNSS systems: GPS, Galileo, GLONASS and BeiDou. Owing to the dual-frequency RF front-end architecture, either GLONASS or BeiDou can be processed concurrently with GPS and Galileo signals providing reception of three GNSS systems. By default the M8 receivers are configured for concurrent GPS and GLONASS, including SBAS and QZSS reception. If power consumption is a key factor, the receiver should be configured for a single GNSS operation using GPS, Galileo, GLONASS or BeiDou and disabling QZSS and SBAS. The module can be configured to receive any single GNSS constellation or within the set of permissible combinations shown below.

GPS	Galileo	GLONASS	BeiDou
•	•	–	–
•	•	•	–
•	•	–	•
•	–	•	–
•	–	–	•
–	•	•	–
–	•	–	•
–	–	•	•

Table 2 Permissible GNSS combinations (• = enabled)

- The SBAS and QZSS augmentation systems can be enabled only if GPS operation is configured.
- Galileo is not enabled as the default configuration.

1.5.1 GPS

The NEO-M8 positioning modules are designed to receive and track the L1C/A signals provided at 1575.42 MHz by the global positioning system (GPS).

1.5.2 GLONASS

The NEO-M8 modules can receive and process the GLONASS satellite system as an alternative to the US-based global positioning system (GPS). u-blox NEO-M8 positioning modules are designed to receive and track the L1OF signals GLONASS provides at 1602 MHz + $k \cdot 562.5$ kHz, where k is the satellite's frequency channel number ($k = -7, \dots, 5, 6$). The ability to receive and track GLONASS L1OF satellite signals allows design of GLONASS receivers where required by regulations.



To take advantage of GPS and GLONASS, dedicated hardware preparation must be made during the design-in phase. See the NEO-8Q / NEO-M8 Hardware integration manual [1] for u-blox design recommendations.

1.5.3 BeiDou

The NEO-M8 modules can receive and process the B1I signals broadcast at 1561.098 MHz from the BeiDou Navigation Satellite System. The ability to receive and track BeiDou signals in conjunction with another constellation results in higher coverage, improved reliability and better accuracy. Currently, BeiDou is not fully operational globally and provides Chinese regional coverage only. Global coverage is scheduled for 2020.

1.5.4 Galileo

The NEO-M8 positioning modules can receive and track the E1-B/C signals centered on the GPS L1 frequency band. GPS and Galileo signals can be processed concurrently together with either BeiDou or GLONASS signals, enhancing coverage, reliability and accuracy. The SAR return link message (RLM) parameters for both short and long versions are decoded by the receiver and made available to users via UBX proprietary messages.

-  Galileo has been implemented according to ICD release 1.2 (November 2015) and verified with live signals from the Galileo in-orbit validation campaign. Since the Galileo satellite system has not yet reached Initial (IOC) nor Full Operational Capability (FOC), changes to the Galileo signal specification (OS SIS ICD) remain theoretically possible. u-blox therefore recommends using flash memory in designs utilizing Galileo signals to allow for a FW update in the unlikely event of a change to the Galileo signal specification (OS SIS ICD).
-  Galileo reception is by default disabled, but can be enabled by sending a configuration message (UBX-CFG-GNSS) to the receiver. See the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2] for more information.

1.6 Assisted GNSS (A-GNSS)

Supply of GNSS receiver assistance information, such as ephemeris, almanac, rough user position and time, will reduce the time to first fix significantly and improve acquisition sensitivity. The NEO-M8N, NEO-M8Q, and NEO-M8M modules support the u-blox AssistNow Online and AssistNow Offline A-GNSS services, support AssistNow Autonomous, and are OMA SUPL compliant.

1.6.1 AssistNow™ Online


With AssistNow Online, an internet-connected host downloads assistance data from the u-blox AssistNow Online service to the receiver at system start-up. The Multi-GNSS Assistance (MGA) service is an HTTP protocol-based network operator-independent service.

Supplying assistance information, such as ephemeris, almanac, a rough last position and time, can reduce the time to first fix significantly and improve acquisition sensitivity.

-  The AssistNow Online service provides data for GPS, GLONASS, BeiDou, Galileo and QZSS

1.6.2 AssistNow™ Offline

With the AssistNow Offline service, users can download long-term orbit data over the internet at their convenience. The orbit data can be stored in the GNSS receiver's SQL flash memory or, alternatively, within the memory of the application processor. The function requires no connectivity at system start-up, enabling a position fix within seconds, even when no network is available. AssistNow Offline offers augmentation for up to 35 days.


 AssistNow Offline service provides data for GPS and GLONASS only, BeiDou and Galileo are not currently supported.

1.6.3 AssistNow™ Autonomous

AssistNow Autonomous provides aiding information without the need for a host or external network connection. Based on previous broadcast satellite ephemeris data downloaded to and stored by the GNSS receiver, AssistNow Autonomous automatically generates accurate predictions of satellite orbital data ("AssistNow Autonomous data") that is usable for future GNSS position fixes. The concept capitalizes on the periodic nature of GNSS satellites; by capturing strategic ephemeris data at specific times of the day. The flash memory-based NEO-M8N module can predict accurate satellite ephemeris for up to six days after initial reception. The ROM-based NEO-M8M/Q modules can use only GPS satellites with a prediction time of up to three days.

u-blox AssistNow Autonomous benefits are:

- Faster fix in situations where satellite signals are weak
- No connectivity required
- Compatible with AssistNow Online (can work stand-alone, or in tandem with AssistNow Online service)
- No integration effort; calculations are done in the background, transparent to the user

 For more details on A-GNSS, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.7 Augmentation systems

1.7.1 Satellite-based augmentation system (SBAS)

The u-blox NEO-M8 modules support reception of SBAS broadcast signals. These systems supplement GNSS data with additional regional or wide area GPS augmentation data. The system broadcasts range correction and integrity information via satellite which can be used by GNSS receivers to improve resulting precision. SBAS satellites can be used as additional satellites for ranging (navigation), further enhancing availability. The following SBAS types are supported: GAGAN, WAAS, EGNOS and MSAS.

 For more details, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].


1.7.2 QZSS

The Quasi-Zenith Satellite System (QZSS) is a regional navigation satellite system that transmits additional GPS L1 C/A signals for the Pacific region covering Japan and Australia. NEO-M8 positioning modules are able to receive and track these signals concurrently with GPS signals, resulting in better availability especially under challenging signal conditions, e.g. in urban canyons. The L1- SAIF signal provided by QZSS can be enabled for reception via a GNSS configuration message.

1.7.3 IMES

The Japanese Indoor Messaging System (IMES) system is used for indoor position reporting using low-power transmitters which broadcast a GPS-like signal. NEO-M8 modules can be configured to receive and demodulate the signal to provide an in-door location estimate.

 This service is authorized and available only in Japan.

 IMES reception is disabled by default.

1.7.4 Differential GPS (D-GPS)

u-blox receivers support Differential GPS (D-GPS) data according to RTCM specification 10402.3 [4]: The use of D-GPS improves GPS position accuracy. The RTCM implementation supports the following RTCM 2.3 messages.

Message type	Description
1	Differential GPS corrections
2	Delta differential GPS corrections
3	GPS reference station parameters
9	GPS partial correction set

Table 3: Supported RTCM 2.3 messages

 RTCM corrections cannot be used together with SBAS.


 For more details, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.8 Broadcast navigation data and satellite signal measurements

The NEO-M8 modules can output all the GNSS broadcast data upon reception from tracked satellites. This includes all the supported GNSS signals plus the augmentation services SBAS, QZSS and IMES. The receiver also makes available the tracked satellite signal information, that is, raw code phase and Doppler measurements in a form aligned to the ETSI mobile cellular location services protocol (RRLP) [6]. For more details, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.9 Odometer

The odometer function provides information on travelled ground distance (in meters) based on the position and Doppler-based velocity output from the navigation solution. For each computed distance since the last odometer reset, the odometer estimates a 1-sigma accuracy value. The total cumulative ground distance is maintained and saved in the BBR memory.

 The odometer feature is disabled by default. For more details, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.10 Data logging (NEO-M8N)

The NEO-M8N module can be used in data logging applications. The data logging feature enables continuous storage of position, velocity and time information to an onboard SQL flash memory (at least 16 Mbit). It can also log the distance from the odometer. The information can be downloaded from the receiver later for further analysis or for conversion to a mapping tool. For more information see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.11 Geofencing

The u-blox NEO-M8 modules support up to four circular Geofencing areas defined on the Earth's surface using a 2D model. Geofencing is active when at least one Geofence is defined, the current status can be found by polling the receiver. A GPIO pin can be nominated to indicate status to, for example, wake up a host on activation.

1.12 Message integrity protection

The NEO-M8 modules provide a function to detect third party interference with the UBX message stream sent from receiver to host. The security mechanism "signs" nominated messages via a subsequent UBX message. This message signature is then compared with one generated by the host to determine if the message data has been altered. The signature algorithm seed can use one fixed secret ID key set by eFuse in production and a dynamic ID key set by the host, enabling users to detect "man-in-the-middle" style attacks.

1.13 Spoofing detection

Spoofing is a process whereby a malicious third party tries to control the reported position via a "fake" GNSS broadcast signal. This may result in the form of reporting incorrect position, velocity or time. To combat against this, NEO-M8 modules include spoofing detection measures to alert the host when signals appear to be suspicious. The receiver combines a number of checks on the received signals looking for inconsistencies across several parameters.

 This feature does not guarantee detecting all spoofing attacks.

1.14 EXTINT: External interrupt

EXTINT is an external interrupt pin with fixed input voltage thresholds with respect to **VCC**. It can be used for control of the receiver or for aiding.

For more information about how to implement and configure these features, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2] and the NEO-8Q / NEO-M8 Hardware integration manual [1].

1.14.1 Pin control

The pin control feature allows overriding the automatic active/inactive cycle of power save mode. The state of the receiver can be controlled through the **EXTINT** pin.

The receiver can also be forced OFF using **EXTINT** when power save mode is not active.

1.14.2 Aiding

The **EXTINT** pin can be used to supply time or frequency aiding data to the receiver.

For time aiding, hardware time synchronization can be achieved by connecting an accurate time pulse to the **EXTINT** pin.

Frequency aiding can be implemented by connecting a periodic rectangular signal with a frequency up to 500 kHz and arbitrary duty cycle (low/high phase duration must not be shorter than 50 ns) to the **EXTINT** pin. Provide the applied frequency value to the receiver using UBX messages.

1.15 TIMEPULSE

A configurable time pulse signal is available with the NEO-M8N, NEO-M8Q, and NEO-M8M modules.

The **TIMEPULSE** output generates pulse trains synchronized with GPS or UTC time grid with intervals configurable over a wide frequency range. Thus it may be used as a low frequency time synchronization pulse or as a high frequency reference signal.

By default the time pulse signal is configured to 1 pulse per second. For more information, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.16 Protocols and interfaces

Protocol	Type
NMEA 0183, version 4.0 (V2.1, V2.3 or V4.1 configurable)	Input/output, ASCII
UBX	Input/output, binary, u-blox proprietary
RTCM	Input message, 1, 2, 3, 9

Table 4: Available protocols

All protocols are available on UART, USB, DDC (I2C-compliant) and SPI. For specification of the various protocols, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.17 Interfaces

A number of interfaces are provided either for data communication or memory access. The embedded firmware uses these interfaces according to their respective protocol specifications.

1.17.1 UART

The NEO-M8 modules include one UART interface, which can be used for communication to a host. It supports configurable baud rates. For supported baud rates, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].



Designs must allow access to the UART and the **SAFEBOOT_N** function pin for future service, updates and reconfiguration.

1.17.2 USB

A USB interface, which is compatible to USB version 2.0 FS (full speed, 12 Mbit/s), can be used for communication as an alternative to the UART. The pull-up resistor on pin USB_DP is integrated to signal a full-speed device to the host. The VDD_USB pin supplies the USB interface. The u-blox USB (CDC-ACM) driver supports Windows Vista and Windows 7 and 8 operating systems. A separate driver (CDC-ACM) is not required for Windows 10 which has a built-in USB-serial driver. However, plugging initially into an internet-connected Windows 10 PC will download the u-blox combined sensor and VCP driver package.



USB drivers can be downloaded from the u-blox web site, www.u-blox.com.

1.17.3 SPI

The SPI interface is designed to allow communication to a host CPU. The interface can be operated in slave mode only. The maximum transfer rate using SPI is 125 kB/s and the maximum SPI clock frequency is 5.5 MHz. Note that SPI is not available in the default configuration because its pins are shared with the UART and DDC interfaces. The SPI interface can be enabled by connecting **D_SEL** (pin 2) to ground (see section 3.1).

1.17.4 Display data channel (DDC)

An I2C-compliant DDC interface is available for communication with an external host CPU or u-blox cellular modules. The interface can be operated in slave mode only. The DDC protocol and electrical interface are fully compatible with the fast-mode of the I2C industry standard. Since the maximum SCL clock frequency is 400 kHz, the maximum transfer rate is 400 kbit/s.

1.18 Clock generation

1.18.1 Oscillators

NEO-M8 GNSS modules are available in TCXO and crystal versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

Oscillators used on NEO-M8 modules are carefully selected and screened for stability and against frequency perturbations across the full operating range (–40 °C to +85 °C).

The careful selection and qualification of critical parts, such as GNSS oscillators, has resulted in u-blox modules being the most reliable positioning modules in the industry, particularly in challenging conditions.

1.18.2 Real-time clock (RTC)


The RTC is driven by a 32 kHz oscillator using an RTC crystal. If the main supply voltage fails, and a battery is connected to **V_BCKP**, parts of the receiver switch off, but the RTC still runs providing a timing reference for the receiver. This operating mode is called hardware backup mode, which enables all relevant data to be saved in the backup RAM to allow a hot or warm start later.

1.19 Power management

u-blox M8 technology offers a power-optimized architecture with built-in autonomous power saving functions to minimize power consumption at any given time. Furthermore, the receiver can be used in two operating modes: continuous mode for best performance or power save mode for optimized power consumption, respectively.

1.19.1 DC-DC converter

The NEO-M8 modules integrate a DC-DC converter, allowing reduced power consumption especially when using a main supply voltage above 2.5 V.

 For more information, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

1.19.2 Power mode setup

u-blox M8 modules can be configured to run in either continuous or a choice of power save mode configurations. A template of power mode settings can be used to easily select typical power mode setups to cover the majority of users' requirements.

For specific power saving applications the user has the option to fully configure via the power save mode configuration. For more information, see section 1.19.4.

The u-blox M8 modules' power mode setup offers a choice of continuous operation and preset power save mode configurations.

- Continuous (default) mode for best GNSS performance vs. power consumption
- Continuous with no compromise in power consumption
- A 1 Hz cyclic tracking mode for aggressive power reduction

- Choice of 2 or 4 Hz⁸ cyclic tracking modes for typical wearable applications
- ON/OFF interval mode

1.19.3 Continuous mode

Continuous mode uses the acquisition engine at full performance resulting in the shortest possible TTFF and the highest sensitivity. It searches for all possible satellites until the Almanac is completely downloaded. The receiver then switches to the tracking engine to lower power consumption.

Thus, a lower tracking current consumption level will be achieved when:

- A valid GNSS position is obtained
- The entire Almanac has been downloaded
- The Ephemeris for each satellite in view is valid

1.19.4 Power save mode

For specific power saving applications outside the typical preset power mode setups, users can configure a tailored power save mode.

The power save mode provides two dedicated methods, ON/OFF and cyclic tracking, that reduce average current consumption in different ways to match the needs of the specific application. These operations can be set by using a specific UBX message.



For more information about power management strategies, see the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2].

1.20 Antenna

NEO-M8 modules are designed for use with passive⁹ and active¹⁰ antennas.

Parameter	Specification	
Antenna type	Passive and active antenna	
Active antenna recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 ¹¹ dB / 30 ¹² dB
	Maximum noise figure	1.5 dB

Table 5: Antenna specifications for NEO-M8 modules

1.21 LNA_EN

The LNA_EN pin provides optional control for switching off power to an active antenna or separate LNA. This facility is provided to help minimize power consumption in power save mode operation. See the NEO-8Q / NEO-M8 Hardware integration manual [1] for more information.

⁸ Single GNSS constellation configuration only.

⁹ For integration of NEO-M8 modules with Cellular products, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

¹⁰ For using active antennas with NEO-M8 modules, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

¹¹ NEO-M8M

¹² NEO-M8N/Q

2 Pin definition

2.1 Pin assignment




PIO = Peripheral Input Output

Figure 2: Pin assignment

No.	Name	PIO no.	I/O	Description
1	SAFEBOOT_N	-	I	SAFEBOOT_N (for future service, updates and reconfiguration, leave OPEN)
2	D_SEL	-	I	Interface select
3	TIMEPULSE	11	O	Time pulse (1PPS)
4	EXTINT	13	I	External interrupt pin
5	USB_DM	-	I/O	USB data
6	USB_DP	-	I/O	USB data
7	VDD_USB	-	I	USB supply
8	RESET_N	-	I	RESET_N
9	VCC_RF	-	O	Output voltage RF section
10	GND	-	I	Ground
11	RF_IN	-	I	GNSS signal input
12	GND	-	I	Ground
13	GND	-	I	Ground
14	LNA_EN / Reserved	16	O	LNA_EN (NEO-M8N/Q): Antenna/LNA control Reserved (NEO-M8M): Reserved
15	Reserved	-	-	Reserved
16	Reserved	-	-	Reserved
17	Reserved	-	-	Reserved
18	SDA / SPI CS_N	9	I/O	DDC data if D_SEL = 1 (or open) SPI chip select if D_SEL = 0
19	SCL / SPI CLK	8	I/O	DDC clock if D_SEL = 1 (or open) SPI clock if D_SEL = 0
20	TXD / SPI MISO	6	O	Serial port if D_SEL = 1 (or open) SPI MISO if D_SEL = 0
21	RXD / SPI MOSI	7	I	Serial port if D_SEL = 1 (or open) SPI MOSI if D_SEL = 0
22	V_BCKP	-	I	Backup voltage supply
23	VCC	-	I	Supply voltage
24	GND	-	I	Ground

Table 6: Pinout of NEO-M8

 Pins designated as “Reserved” should not be used. For more information about pinouts, see the NEO-8Q / NEO-M8 Hardware integration manual [\[1\]](#).

2.2 Pin name changes

Selected pin names have been updated to agree with a common naming convention across u-blox modules. The pins have not changed their operation and are the same physical hardware but with updated names. The table below lists the pins that have a changed name along with their old and new names.

No.	Previous name	New name
1	Reserved	SAFEBOOT_N
14	ANT_ON	LNA_EN
20	TxD SPI MISO	TXD / SPI MISO
21	RxD SPI MOSI	RXD / SPI MOSI

Table 7: Pin name changes

3 Configuration management

Configuration settings can be modified with UBX configuration messages. The modified settings remain effective until power-down or reset. If these settings have been stored in battery-backup RAM, the modified configuration will be retained as long as the backup battery supply is not interrupted.

With the NEO-M8, configuration settings modified with UBX configuration messages can be saved permanently. In this case, the modified settings remain effective even after power-down and do not require backup battery supply.

3.1 Interface selection (D_SEL)

At startup, pin 2 (**D_SEL**) determines which data interfaces are used for communication. If **D_SEL** is set high or left open, UART and DDC become available. If **D_SEL** is set low, that is, connected to ground, the NEO-M8 module series can communicate to a host via SPI.

Pin #	D_SEL="1" (left open)	D_SEL="0" (connected to GND)
20	UART TX	SPI MISO
21	UART RX	SPI MOSI
19	DDC SCL	SPI CLK
18	DDC SDA	SPI CS_N

Table 8: Data interface selection by D_SEL

4 Electrical specification

The limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the characteristics sections of the specification is not implied. Exposure to these limits for extended periods may affect device reliability.

Where application information is given, it is advisory only and does not form part of the specification. For more information, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

4.1 Absolute maximum rating

Parameter	Symbol	Condition	Min	Max	Units
Power supply voltage	VCC		-0.5	3.6	V
Backup battery voltage	V_BCKP		-0.5	3.6	V
USB supply voltage	VDD_USB		-0.5	3.6	V
Input pin voltage	V _{IN}	If VCC < 3.1V	-0.5	VCC+0.5	V
		If VCC > 3.1V	-0.5	3.6	V
	V _{IN_USB}		-0.5	VDD_USB	V
DC current through any digital I/O pin (except supplies)	I _{PIN}			10	mA
VCC_RF output current	ICC_RF			100	mA
Input power at RF_IN	P _{RFIN}	source impedance = 50 Ω, continuous wave		15	dBm
Storage temperature	T _{STG}		-40	85	°C

Table 9: Absolute maximum ratings

⚠ Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in Table 9, must be limited to values within the specified boundaries by using appropriate protection diodes.

4.2 Operating conditions

All specifications are at an ambient temperature of +25 °C. Extreme operating temperatures can significantly impact specification values. Applications operating near the temperature limits should be tested to ensure the specification.

Parameter	Symbol	Min	Typical	Max	Units	Condition
Power supply voltage NEO-M8N/Q	VCC	2.7	3.0	3.6	V	
Power supply voltage NEO-M8M	VCC	1.65	1.8, 3.0	3.6	V	
Supply voltage USB	VDDUSB	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	1.4		3.6	V	
Backup battery current	I_BCKP		15		μA	V_BCKP = 1.8 V, VCC = 0 V
SW backup current	I_SWBCKP		30		μA	VCC = 3 V
Input pin voltage range	V _{IN}	0		VCC	V	
Digital IO pin low level input voltage	V _{IL}	0		0.2*VCC	V	
Digital IO pin high level input voltage	V _{IH}	0.7*VCC		VCC	V	
Digital IO pin low level output voltage	V _{OL}			0.4	V	I _{OL} = 4mA

Parameter	Symbol	Min	Typical	Max	Units	Condition
Digital IO pin high level output voltage	V _{OH}	VCC-0.4			V	I _{OH} = 4mA
Pull-up resistor for RESET_N (internal)	R _{PU}		11		kΩ	
USB_DM, USB_DP	V _{INU}	Compatible with USB with 27 Ω series resistance				
VCC_RF voltage	VCC_RF		VCC-0.1		V	
VCC_RF output current	ICC_RF			50	mA	
Receiver chain noise figure ¹³	NF _{TOT}		2.0		dB	
Operating temperature	T _{OPR}	-40		85	°C	

Table 10: Operating conditions

Operation beyond the specified operating conditions can affect device reliability.

4.3 Indicative current requirements

Table 11 lists examples of the total system supply current for a possible application.

Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

Parameter	Symbol	Module	Typical GPS & GLONASS	Typical GPS	Max	Units	Condition
Max. supply current ¹⁴	I _{CCP}	All			67	mA	
Average supply current ^{15, 16}	I _{CC} Acquisition ¹⁷	NEO-M8N	32	25		mA	Estimated at 3 V
		NEO-M8M	24	19		mA	
		NEO-M8Q	30	24		mA	
	I _{CC} Tracking (continuous mode)	NEO-M8N	30	23		mA	Estimated at 3 V
		NEO-M8M	21	17		mA	
		NEO-M8Q	28	23		mA	
	I _{CC} Tracking (Power Save mode / 1 Hz)	NEO-M8N	13	12		mA	Estimated at 3 V
		NEO-M8M	5.3	4.8		mA	
		NEO-M8Q	11.5	11.1		mA	

Table 11: Indicative power requirements at 3.0 V

For more information about power requirements, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

For more information on how to noticeably reduce current consumption, see the Power Management Application Note [5].

¹³ Only valid for the GPS band.

¹⁴ Use this figure to dimension maximum current capability of power supply. Measure this parameter with 1 Hz bandwidth.

¹⁵ Use this figure to determine required battery capacity.

¹⁶ Simulated GNSS constellation using power levels of -130 dBm. VCC = 3.0 V.

¹⁷ Average current from start-up until the first fix.

4.4 SPI timing diagrams

To avoid incorrect operation of the SPI, the user needs to comply with certain timing conditions. Consider the following signals for timing constraints:

Symbol	Description
SPI CS_N (SS_N)	Slave select signal
SPI CLK (SCK)	Slave clock signal

Table 12: Symbol description

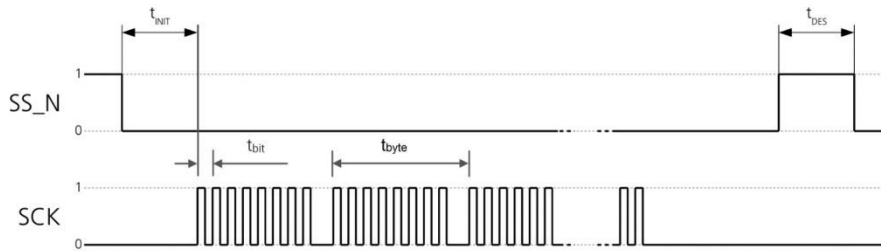


Figure 3: SPI timing diagram

4.4.1 Timing recommendations

The recommendations below are based on a firmware running from flash memory.

Parameter	Description	Recommendation
t_{INIT}	Minimum initialization time	10 μ s
t_{DES}	Deselect time	1 ms
t_{bit}	Minimum bit time	180 ns (5.5 MHz maximum bit frequency)
t_{byte}	Minimum byte period	8 μ s (125 kHz maximum byte frequency)

Table 13: SPI timing recommendations

- The values in Table 13 result from the requirement of an error-free transmission. By allowing just a few errors and disabling the glitch filter, the bit rate can be increased considerably.

4.5 DDC timing diagrams

The DDC interface is I²C Fast Mode compliant. For timing parameters, consult the I²C standard.

- The maximum bit rate is 400 kbit/s. The interface stretches the clock when slowed down when serving interrupts, so real bit rates may be slightly lower.

5 Mechanical specifications

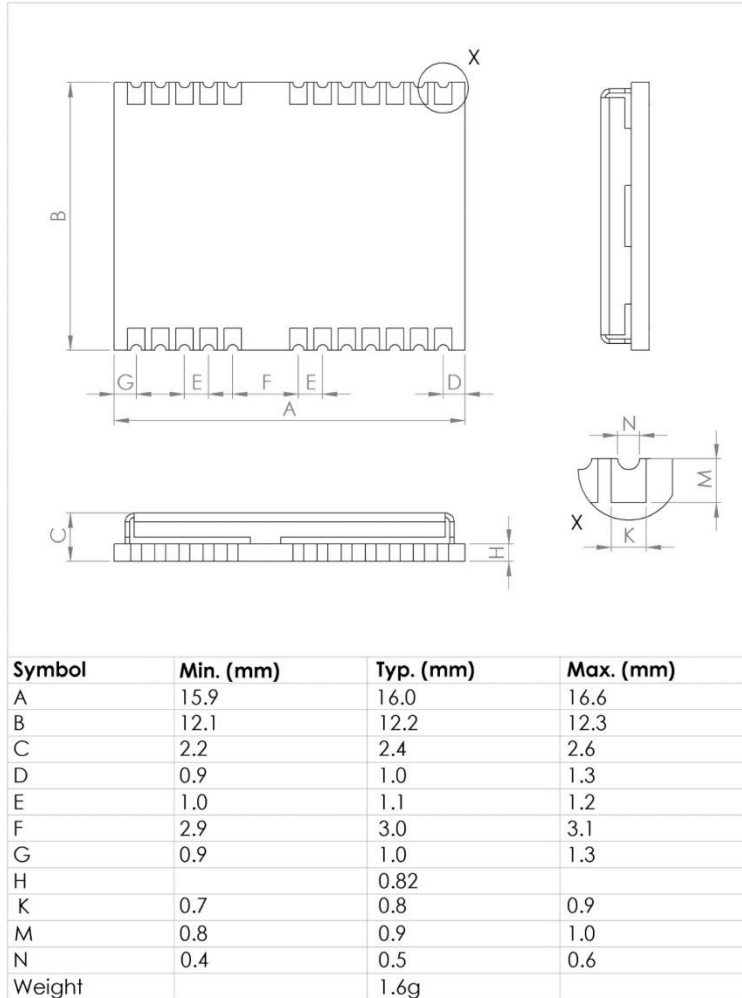


Figure 4: Dimensions

For information about the paste mask and footprint, see the NEO-8Q / NEO-M8 Hardware integration manual [1].

6 Reliability tests and approvals

6.1 Reliability tests



The NEO-M8N, NEO-M8Q, and NEO-M8M modules are based on AEC-Q100 qualified GNSS chips.

Tests for product family qualifications are according to ISO 16750 "Road vehicles – environmental conditions and testing for electrical and electronic equipment", and appropriate standards.

6.2 Approvals



The NEO-M8N, NEO-M8Q, and NEO-M8M modules comply with the Directives 2011/65/EU and 2015/863/EU of the European Parliament and the Council on the Restriction of Use of certain Hazardous Substances (RoHS).

7 Product handling and soldering

7.1 Packaging

To enable efficient production, production lot set-up and tear-down, the NEO-M8 GNSS modules are delivered as hermetically sealed, reeled tapes. For more information, see the u-blox Package Information Guide [3].

7.1.1 Reels

The NEO-M8 GNSS modules are deliverable in quantities of 250 pcs on a reel. The NEO-M8 modules are shipped on reel type B, as specified in the u-blox Package Information Guide [3].

7.1.2 Tapes

The dimensions and orientations of the tapes for NEO-M8 modules are specified in Figure 5.

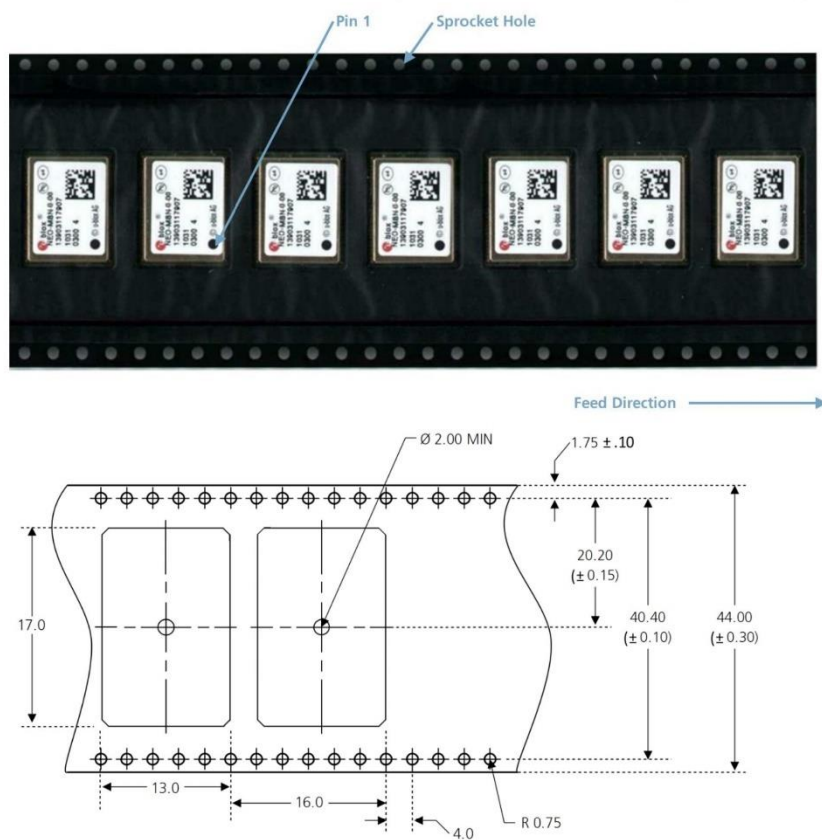



Figure 5: Dimensions and orientation for NEO-M8 modules on tape


7.2 Shipment, storage and handling

For important information regarding shipment, storage and handling, see the u-blox Package Information Guide [3].

7.2.1 Moisture sensitivity levels

The moisture sensitivity level (MSL) relates to the packaging and handling precautions required. The NEO-M8 modules are rated at MSL level 4.


 For MSL standard see IPC/JEDEC J-STD-020, which can be downloaded from www.jedec.org.

 For more information regarding MSL, see the u-blox Package Information Guide [3].

7.2.2 Reflow soldering

Reflow profiles are to be selected according u-blox recommendations (see the NEO-8Q / NEO-M8 Hardware Integration Manual [1]).

7.2.3 ESD handling precautions

 NEO-M8 modules are electrostatic sensitive devices (ESD). Observe precautions for handling! Failure to observe these precautions can result in severe damage to the GNSS receiver!

GNSS receivers are electrostatic sensitive devices (ESD) and require special precautions when handling. Exercise particular care when handling patch antennas, due to the risk of electrostatic charges. In addition to standard ESD safety practices, take the following measures into account whenever handling the receiver:

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device.
- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10 pF, coax cable ~50-80 pF/m, soldering iron).
- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in a non-ESD protected work area, implement proper ESD protection measures in the design.
- When soldering RF connectors and patch antennas to the receiver's RF pin, make sure to use an ESD safe soldering iron (tip).



8 Default messages

Interface	Settings
UART output	9600 baud, 8 bits, no parity bit, 1 stop bit. Configured to transmit both NMEA and UBX protocols, but only the following NMEA (and no UBX) messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT
USB output	Configured to transmit both NMEA and UBX protocols, but only the following NMEA (and no UBX) messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT USB power mode: bus-powered
UART input	9600 baud, 8 bits, no parity bit, 1 stop bit, autobauding disabled. Automatically accepts following protocols without need of explicit configuration: UBX, NMEA, RTCM The GNSS receiver supports interleaved UBX and NMEA messages.
USB input	Automatically accepts following protocols without need of explicit configuration: UBX, NMEA The GPS receiver supports interleaved UBX and NMEA messages. USB power mode: bus-powered
DDC	Fully compatible with the I2C industry standard, available for communication with an external host CPU or u-blox cellular modules, operated in slave mode only. Default messages activated. NMEA and UBX are enabled as input messages, only NMEA as output messages. Maximum bit rate 400 kbit/s.
SPI	Allow communication to a host CPU, operated in slave mode only. Default messages activated. SPI is not available in the default configuration.
TIMEPULSE (1 Hz Nav)	1 pulse per second, synchronized at rising edge, pulse length 100 ms

Table 14: Default messages



Refer to the u-blox 8 / u-blox M8 Receiver Description including Protocol Specification [2] for information about other settings.

9 Labeling and ordering information

9.1 Product labeling

The labeling of u-blox NEO-M8 GNSS modules includes important product information. The location of the NEO-M8 product type number is shown in Figure 6.

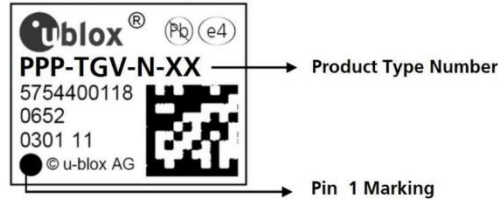


Figure 6: Location of product type number on u-blox NEO-M8 module label

9.2 Explanation of codes

Three different product code formats are used. The **Product name** is used in documentation such as this data sheet and identifies all u-blox M8 products, independent of packaging and quality grade. The **Ordering code** includes options and quality, while the **Type number** includes the hardware and firmware versions. Table 15 shows the structure of these three different formats.

Format	Structure
Product name	PPP-TGV
Ordering code	PPP-TGV-N
Type number	PPP-TGV-N-XX

Table 15: Product code formats

The parts of the product code are explained in Table 16.

Code	Meaning	Example
PPP	Product family	NEO
TG	Platform	M8 = u-blox M8
V	Variant	Function set (A-Z), T = Timing, R = DR, etc.
N	Option / Quality grade	Describes standardized functional element or quality grade 0 = Default variant, A = Automotive
XX	Product detail	Describes product details or options such as hard- and software revision, cable length, etc.

Table 16: Part identification code

9.3 Ordering codes

Ordering code	Product
NEO-M8M-0	u-blox M8 Concurrent GNSS LCC Module, crystal, ROM, 12.2x16 mm, 250 pcs/reel
NEO-M8N-0	u-blox M8 Concurrent GNSS LCC Module, TCXO, flash, SAW, LNA, 12.2x16 mm, 250 pcs/reel
NEO-M8Q-0	u-blox M8 Concurrent GNSS LCC Module, TCXO, ROM, SAW, LNA, 12.2x16 mm, 250 pcs/reel

Table 17: Product ordering codes for professional grade modules

Product changes affecting form, fit or function are documented by u-blox. For a list of Product Change Notifications (PCNs), see our website.

Appendix

A Glossary

Abbreviation	Definition
AEC	Automotive Electronics Council
BBR	Battery Backed RAM
DDC	Display Data Channel
EGNOS	European Geostationary Navigation Overlay Service
ESD	Electrostatic Sensitive Device*
FOC	Full Operational Capability
GAGAN	GPS Aided GEO Augmented Navigation
GLONASS	GLobal Navigation Satellite System (Russian)
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPS	Global Positioning System
IMES	Indoor MESSaging System
I2C	Inter-Integrated Circuit
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LCC	Leadless Chip Carrier
LCS	LoCation Services (protocol)
LNA	Low Noise Amplifier
MSAS	MTSAT Satellite Augmentation System
MSL	Moisture Sensitivity Level
NMEA	National Marine Electronics Association
PPP	Point-to-Point Protocol* / Precise Point Positioning*
PCB	Printed Circuit Board
PCN	Product Change Notification
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RLM	Return Link Message
RRLP	Radio Resource LCS Protocol
RTC	Real Time Clock
RTCM	Radio Technical Commission for Maritime Services
SAW	Surface Acoustic Wave
SBAS	Satellite-Based Augmentation System
SCL	Serial Clock
SMD	Solder Mask Defined
SUPL	Secure User Plane Location
TCXO	Temperature-Compensated Crystal Oscillator
TTFF	Time-To-First-Fix
UART	Universal Asynchronous Receiver/Transmitter
UTC	Coordinated Universal Time
WAAS	Wide Area Augmentation System

Table 18: Explanation of the abbreviations and terms used



Contact

For complete contact information, visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 703 483 3180

E-mail: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 408 573 3640

E-mail: info_us@u-blox.com

Technical Support:

Phone: +1 703 483 3185

E-mail: support@u-blox.com

Headquarters

Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44

E-mail: info@u-blox.com

Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811

E-mail: info_ap@u-blox.com

Support: support_ap@u-blox.com

Regional Office Australia:

Phone: +61 2 8448 2016

E-mail: info_au@u-blox.com

Support: support_ap@u-blox.com

Regional Office China (Beijing):

Phone: +86 10 68 133 545

E-mail: info_cn@u-blox.com

Support: support_cn@u-blox.com

Regional Office China (Chongqing):

Phone: +86 23 6815 1588

E-mail: info_cn@u-blox.com

Support: support_cn@u-blox.com

Regional Office China (Shanghai):

Phone: +86 21 6090 4832

E-mail: info_cn@u-blox.com

Support: support_cn@u-blox.com

Regional Office China (Shenzhen):

Phone: +86 755 8627 1083

E-mail: info_cn@u-blox.com

Support: support_cn@u-blox.com

Regional Office India:

Phone: +91 80 405 092 00

E-mail: info_in@u-blox.com

Support: support_in@u-blox.com

Regional Office Japan (Osaka):

Phone: +81 6 6941 3660

E-mail: info_jp@u-blox.com

Support: support_jp@u-blox.com

Regional Office Japan (Tokyo):

Phone: +81 3 5775 3850

E-mail: info_jp@u-blox.com

Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861

E-mail: info_kr@u-blox.com

Support: support_kr@u-blox.com

Regional Office Taiwan:

Phone: +886 2 2657 1090

E-mail: info_tw@u-blox.com

Support: support_tw@u-blox.com